

Anne Rantamäki

**Tietokantapohjaisen web-sovelluksen suunnittelu ja toteutus atk-laitetietojen hallintaan**

Opinnäytetyö  
Kevät 2011  
Tekniikan yksikkö  
Tietojenkäsittelyn koulutusohjelma,  
Sovellustuotannon suuntautumisvaihtoehto



## SEINÄJOEN AMMATTIKORKEAKOULU

**Opinnäytetyön tiivistelmä**

Koulutusyksikkö:	Tekniikan yksikkö		
Koulutusohjelma:	Tietojenkäsittelyn koulutusohjelma		
Suuntautumisvaihtoehto:	Sovellustuotannon suuntautumisvaihtoehto		
Tekijä:	Anne Rantamäki		
Työn nimi:	Tietokantapohjaisen web-sovelluksen suunnittelu ja toteutus atk-laitetietojen hallintaan		
Ohjaaja:	Markku Lahti		
Vuosi: 2011	Sivumäärä: 52	Liitteiden lukumäärä:4	

Opinnäytetyön tarkoituksena oli rakentaa relaatiotietokantapohjainen sovellus Seinäjoen ammattikorkeakoulun Sosiaali- ja terveysalan yksikön atk-tukihenkilöille laitetietojen hallintaa varten. Atk-tukihenkilöt pitävät kirjaa kaikista rakennuksessa sijaitsevista atk-laitteista. Aiemmin tietoja hallinnoitiin Excel-taulukolla. Sovelluksen tarkoituksena oli helpottaa laitetietojen hallintaa ja nopeuttaa tiedon etsimistä sekä mahdollistaa laitetietojen helppo lisääminen tietokantaan. Sovelluksen tuli olla helppokäyttöinen ja ulkoasultaan selkeä.

Sovelluksen kehittäminen tapahtui ohjelmistokehittämisen prosessin mukaisesti. Kehittämisessä sovellettiin protoilumallia ja hyödynnettiin UML-mallinnusta. Sovelluksesta valmistettiin prototyyppi, jossa tärkeiden ominaisuuksien toimivuutta kokeiltiin. Prototyyppiä kehitettiin edelleen ja siitä päädyttiin valmiiseen sovellukseen. Sovellus on toteutettu käyttäen Visual Studio 2008 -sovelluskehittäjäsovelletta. Sovellus rakennettiin ASP.NET-menetelmällä käyttäen C#-ohjelmointikieltä. Tietokanta luotiin Microsoft Access 2007 -tietokantatyökalulla.

Sovellus täytti asiakasvaatimukset ja se otettiin käyttöön toimeksiantajalla. Työssä saavutettiin sille asetetut vaatimukset.

## SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

**Thesis abstract**

Faculty:	School of Technology	
Degree programme:	Business Information Technology	
Specialisation:	Software Production	
Author/s:	Anne Rantamäki	
Title of thesis:	Planning and constructing a database based web application to manage the information of the computer hardware	
Supervisor(s):	Markku Lahti	
Year: 2011	Number of pages: 52	Number of appendices: 4

The goal of this thesis was to build a relational database application for Seinäjoki University of Applied Sciences, School Of Health care and Social works IT-Support staff for managing device information. IT-Support staff keeps track of all computer hardware in the building. The information was previously managed with an Excel-table. The purpose of the application was to make it easier to manage computer hardware information and to speed up information searching and also to allow easy device information adding to the database. The application had to be easy to use and have a clear visual appearance.

The development of the application went according to software development process. Prototyping model and UML-modelling was applied in the development. A prototype of the application was made, where the functionality of the important features were tested. The prototype was then developed further and from that resulted the complete application. The application is implemented by using Visual Studio 2008- software development kit. The application was built with ASP.NET-method using C#-programming language. Database was created with Microsoft Access 2007- database application.

The Application fulfilled client demands and it was adopted by the employer. The project achieved the demands set to it.

Keywords: dynamic web-pages, software engineering, relational database

## SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ .....	4
Kuvio- ja taulukkoluettelo.....	6
Esimerkkiluettelo .....	8
Käytetyt termit ja lyhenteet .....	9
1 JOHDANTO .....	10
1.1 Työn tausta .....	10
1.2 Tavoite ja tutkimusongelma .....	11
1.3 Rakenne.....	12
2 RAKENTAMISMETODIT.....	14
2.1 Ohjelmistotuotantoprosessi.....	14
2.2 Vaihejakomallit.....	14
2.3 UML .....	17
2.3.1 Käyttötapauskaavio.....	18
2.3.2 Sekvenssikaavio .....	19
2.3.3 Luokkakaavio .....	19
3 TOIMINNALLINEN MÄÄRITETTLY.....	20
3.1 Asiakasvaatimukset .....	20
3.2 Sovelluksen käyttötapaukset.....	20
3.3 Sovelluksen toiminnot .....	22
3.4 Sovelluksen luokkakaavio .....	23
4 TEKNINEN MÄÄRITTELY.....	25
4.1 Sovelluksen yleiskuvaus .....	25
4.2 Relaatiotietokannat .....	25
4.2.1 Rakenne.....	26
4.2.2 Käsittely .....	26
4.2.3 Eheyssäännöt .....	27
4.2.4 Tietokannan rakenne .....	27

4.3 Sovelluksen rakenne.....	30
4.3.1 Käyttöliittymä.....	30
4.3.2 Haku alasvetovalikoista.....	32
4.3.3 Haku hakusanalla .....	34
4.3.4 Poista käyttäjä/käyttäjän lisätiedot .....	37
4.3.5 Poista laite/laitteen lisätiedot.....	39
4.3.6 Muokkaa laitteen/käyttäjän tietoja .....	41
4.3.7 Lisää käyttäjä .....	42
4.3.8 Lisää laite.....	44
5 SOVELLUKSEN TESTAUS .....	48
6 ARVIOINTI JA JATKOKEHITYS .....	50
LÄHTEET .....	52
LIITTEET .....	53

## Kuvio- ja taulukkoluetelo

Kuvio 1. Konstruktiivinen etenemistapa. ....	12
Kuvio 2. Vesiputousmalli. ....	15
Kuvio 3. Protoilumalli. ....	17
Kuvio 4. Sovelluksen käyttötapaukset. ....	21
Kuvio 5. Sekvenssikaavio tietojen poistamisesta. ....	23
Kuvio 6. Sovelluksen luokkakaavio. ....	24
Kuvio 7. Tietokannan yhteydet. ....	28
Kuvio 8. MasterPage-sivun rautalankamalli. ....	31
Kuvio 9. Sivuston etusivu. ....	32
Kuvio 10. Alasvetovalikot tietojen hakemista varten. ....	32
Kuvio 11. Hakusanahaun lomakekomponentit. ....	34
Kuvio 12. Hakutulossivu, kun haku on tehty käyttäjänimen perusteella. ....	36
Kuvio 13. Varmistusikkuna. ....	39
Kuvio 14. Tulostimen lisätietopaneeli. ....	40
Kuvio 15. Uuden käyttäjän lisäyssivu. ....	43
Kuvio 16. Uuden tietokoneen lisäyssivu. ....	45

Kuvio 17. ValidationSummary.....	46
----------------------------------	----

Kuvio 18. V-malli.....	48
------------------------	----

Taulukko 1. Hakeminen hakusanalla -toiminnon käyttötapauskuvaukset.....	22
---	----

Taulukko 2. Testitapausten käyttäjätietojen muokkaus.....	49
---	----

## Esimerkkiluettelo

Esimerkki 1. Alasvetovalikon täyttäminen käyttäjänimillä.....	33
Esimerkki 2. Uuden ListItem-olion luonti.....	34
Esimerkki 3. Valitun kategorian tarkistaminen if-lauseella.....	35
Esimerkki 4. If-lauseella tarkistetaan, kumpaa hakutapaa on käytetty.....	37
Esimerkki 5. Painikkeen OnClientClick-ominaisuus.....	38
Esimerkki 6. Varmistusikkunan luonti JavaScriptin confirm-funktiolla.....	38
Esimerkki 7. Käyttäjän ja käyttäjälle kuuluvan tietokoneen poistaminen delete-lauseella.....	39
Esimerkki 8. Tekstikenttien, otsikoiden ja painikkeiden asettaminen näkyväksi/näkymättömäksi.....	41
Esimerkki 9. Tulostintietojen muokkaaminen update-lauseella.....	42
Esimerkki 10. Sql-lause tietokoneiden hakemista varten.....	43
Esimerkki 11. SelectedIndexChanged-tapahtuma.....	44
Esimerkki 12. While-rakenne.....	46
Esimerkki 13. Tarkistetaan if-lauseella, onko tekstikenttään syötetty tietoa.....	47



## Käytetyt termit ja lyhenteet

<b>Access</b>	Microsoftin kehittämä ohjelma relaatiotietokantojen luontiin ja hallintaan.
<b>ADO.NET</b>	ActiveX Data Objects on rajapinta, jota käytetään tietokantayhteyksien luontiin Windows-alustalla.
<b>ASP.NET</b>	Active Server Pages on palvelinpuolen ohjelmointimenetelmä, jota käytetään luomaan dynaamisia web-sivuja.
<b>C#</b>	C Sharp on C- ja C++-ohjelmointikielten yhdistelmä.
<b>JavaScript</b>	Web-ympäristön ohjelmointikieli, jolla voidaan lisätä sivuille toiminnallisuutta.
<b>ListItem</b>	Luokka, jonka avulla tiedot kootaan listaan.
<b>SQL</b>	Structured Query Language. Ohjelmointikieli tietokantojen hallintaan.
<b>UML</b>	Unified Modeling Language. Graafinen mallinnuskieli ohjelmistojen rakentamisen eri vaiheisiin.
<b>Visual Studio</b>	Microsoftin kehittämä sovelluskehitin.  (Webopedia. [viitattu 25.1.2011].)

# 1 JOHDANTO

Lähes poikkeuksetta jokaisessa organisaatiossa on erisuuruiset määrät atk-laitteita ja niiden tiedoista pidetään kirjaa. Tämä voi onnistua tallentamalla tiedot esimerkiksi perinteisesti Excel- tai Word-tiedostoon. Suuremmissa organisaatioissa atk-laitteiden määrä on jo niin suuri, että perinteinen ratkaisu ei ole enää käytännöllinen vaihtoehto laitetietojen ylläpitämiseen.

Toimeksiantajan kanssa käydyissä alkukeskusteluissa kävi ilmi, että nykyinen tapa tallentaa ja hakea tietoja Excel-tiedostosta on hidasta ja tietoja ei ole selkeästi jäsennetty. Tietoja oli paljon, joten niiden hallintaa haluttiin helpottaa.

Opinnäytetyössä on tarkoitus rakentaa Seinäjoen ammattikorkeakoulun Sosiaali- ja terveysalan yksikölle atk-laitetietojen hallintasovellus, jonka pohjalla toimii tietokanta. Sovelluksen on tarkoitus helpottaa ja nopeuttaa yksikön atk-tukihenkilöiden päivittäistä työskentelyä.

Opinnäytetyössä käydään läpi sovelluksen suunnittelu ja tekninen toteutus. Sovelluksen toiminnot käydään läpi koodiesimerkein.

## 1.1 Työn tausta

Rakennettava sovellus tulee Seinäjoen ammattikorkeakoulun Sosiaali- ja terveysalan yksikölle atk-tukihenkilöiden käytettäväksi. Atk-tukihenkilöt pitävät kirjaa ammattikorkeakoulun, Sedun sosiaali- ja terveysalan toimipisteen sekä terveysteknologiakeskus Mediwestin tietokoneista, tulostimista, näytöistä, videotykeistä sekä Dvd- ja vhs-laitteista. Tallennettavia tietoja ovat sarjanumerot, mallit, ip-osoitteet, leasing-tiedot, laitteiden sijainnit, käyttäjät ja laitteiden nimet. Kaikki tiedot on tallennettu Excel-taulukkoon, jonka avulla tietoja hallinnoidaan. Excel-taulukkoon tehdään tarvittavat lisäykset, poistot ja muutokset. Taulukossa on kaksi välilehteä, joista toiseen on tallennettu käyttäjät laitetietoineen ja toiseen opetustilojen laitetiedot. Laitetiedot on järjestetty tietokoneen ip-osoitteen mukaan. Muokkaustoimintojen lisäksi Excel-taulukosta suoritetaan kaikki hakutoiminnot. Tämä tapahtuu joko pelkästään taulukkoa silmäilemällä tai käyttämällä ohjelman tarjoamaa Etsi ja

valitse -toimintoa. Excel-taulukko on tallennettu kummankin atk-tukihenkilön työasemalle.

## 1.2 Tavoite ja tutkimusongelma

Tavoitteena on rakentaa Seinäjoen ammattikorkeakoulun Sosiaali- ja terveystieteiden yksikölle tietokantapohjainen web-sovellus helpottamaan laitetietojen hallintaa ja seuranta sekä nopeuttamaan päivittäistä työskentelyä. Sovelluksen avulla laitteista pystyttäisiin tallentamaan tietoja ja suorittamaan erilaisia hakuja. Tarkoituksena on rakentaa relaatiotietokantaa hyödyntävä sovellus, joka on käyttöliittymältään selainpohjainen. Tavoitteena on suunnitella ja toteuttaa ulkoasultaan selkeä, helppokäyttöinen, toimiva ja pitkään palveleva tietojärjestelmä.

Tutkimusongelmana työssä on, miten rakennetaan atk-tukihenkilöiden tarpeita vastaava laitetietojen hallintajärjestelmä. Tutkimusongelmakysymyksiä ovat:

- Miten ohjelmistokehitysprosessi etenee ja mitä kehittämismetodeja on olemassa?
- Miten tietokannan eheys varmistetaan rakenteenmäärittelyvaiheessa?
- Miten sivuston ulkoasu saadaan yhtenäiseksi?
- Miten sovelluksesta saadaan helppokäyttöinen ja käyttäjäystävällinen?

Opinnäytetyötä on lähestytty ohjelmistokehittäjän näkökulmasta. Opinnäytetyötä tehdessä on edetty konstruktivisen tutkimuksen (kuviot 1) mukaan, sillä työssä on luotu uutta käyttäen hyväksi perustutkimuksen tuloksia. Konstruktivisen tutkimuksen ideana on rakentaa uusi todellisuus olemassa olevan tutkimustiedon pohjalta samalla kun ratkaistaan, millaista uutta todellisuutta halutaan rakentaa. Tavoitetila määritetään rakentajien ja päätöksentekijöiden toimesta. Konstruktivisessa tutkimuksessa on kolme päävaihetta: lähtötilanne, toteuttaminen ja tavoitetila. (Järvinen & Järvinen 2000, 105.)



Kuvio 1. Konstruktiivinen etenemistapa.  
Lähde: Järvinen & Järvinen 2000, 105.

Lähtötilanteessa tutustuttiin ohjelmistokehitysprosessiin ja vaihejakomalleihin. Lisäksi tutustuttiin UML-kuvauskieleen ja miten sitä voisi hyödyntää sovelluksen kehittämisessä. Lähtötilanteessa tutustuttiin myös relaatiotietokantojen rakentamis- ja käsittelyteorioihin ja eheyssääntöihin. Lähtötilanteessa tehtiin analyysi nykytilanteesta ja toimeksiantajan kanssa kartoitettiin asiakasvaatimukset. Tavoitetilan saavuttamiseksi toimeksiantajan kanssa keskusteltiin ja neuvoteltiin rakennettavan sovelluksen ominaisuuksista ja toiminnoista. Toteuttamisvaiheessa sovellus rakennettiin valituin menetelmin ja välinein, ja sitä testattiin tekijän ja toimeksiantajan toimesta. Toteutuksen tuloksena syntyi tavoitetilan mukainen sovellus, joka vastasi asiakasvaatimuksia.

### 1.3 Rakenne

Luvussa 2 kuvataan rakentamismetodit, joita hyödyntäen järjestelmää on alettu rakentaa. Lisäksi käydään läpi vaihejakomalleja sekä kerrotaan yleistä tietoa UML-mallinnuksesta ja miten niitä on hyödynnetty opinnäytetyössä.

Luvussa 3 kuvataan asiakasvaatimukset ja selitetään niistä johdetut järjestelmävaatimukset.

Luvussa 4 selitetään miten järjestelmän toiminnot on toteutettu. Toimintojen yhteydessä on käytetty koodiesimerkkejä. Lisäksi kerrotaan yleistä tietoa relaatiotietokannoista.

Luvussa 5 kerrotaan yleistä tietoa järjestelmätestauksesta sekä miten sitä on hyödynnetty järjestelmää testattaessa.

Luvussa 6 arvioidaan työn onnistumista ja asiakasvaatimusten toteutumista. Lisäksi tarkastellaan jatkokehitysnäkymiä.

## 2 RAKENTAMISMETODIT

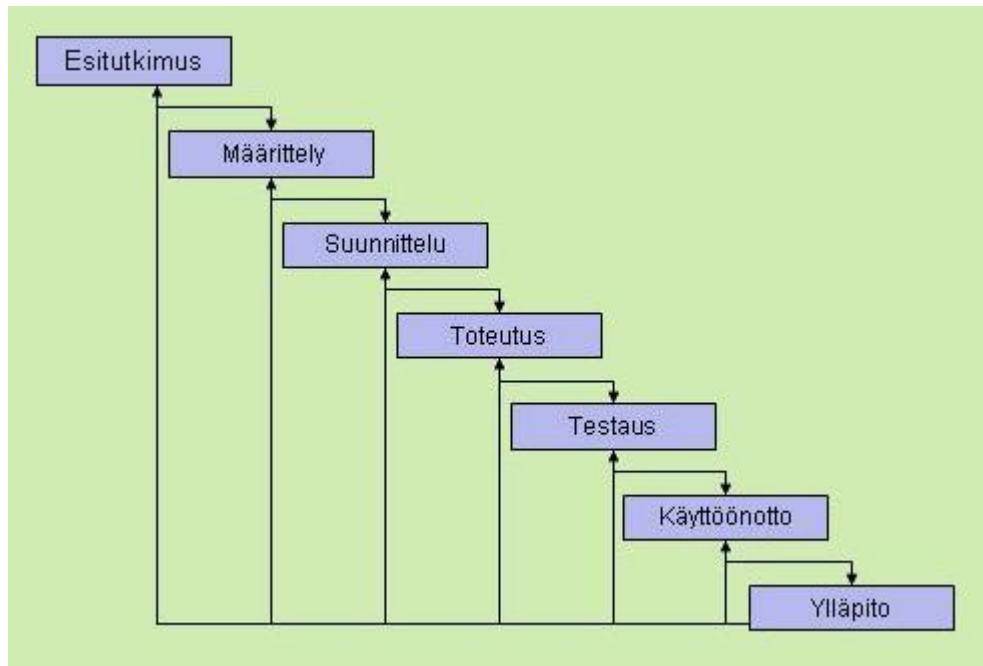
### 2.1 Ohjelmistotuotantoprosessi

Ohjelmistotuotantoprosessilla tarkoitetaan kuvausta käyttäjien tarpeista toimivaksi järjestelmäksi sekä tästä syntyvien määrittelyjen pohjalta itse ohjelmiston tuottamista. Tuotantoprosessin tarkoituksena on päätyä asiakasvaatimuksista toimivaan sovellukseen, jonka asiakas ottaa käyttöönsä. Prosessiin kuuluu esitutkimus, asiakasvaatimusten määrittely, sovelluksen suunnittelu ja toteutus, testaus, käyttöönotto ja ylläpito. Prosessissa on myös tukitoimintoja, joista tärkeimpiä ovat tuotteenhallinta, laadunvarmistus ja dokumentointi (Haikala & Märijärvi 2004, 13, 35).

### 2.2 Vaihejakomallit

Ohjelmistokehityksessä käytetään vaihejakomalleja, joiden avulla ohjelmiston elinkaaresta pystytään erottamaan siihen kuuluvat vaiheet. Elinkaari tarkoittaa aikaa ohjelmiston kehityksen aloittamisesta ohjelmiston poistamiseen käytöstä. Vaihejakomallit voidaan jakaa lineaarisiin ja evolutionäärisiin malleihin. Lineaarinen malli on esimerkiksi vesiputousmalli, jossa on selkeät peräkkäiset vaiheet. Lineaarisissa malleissa edetään suoraviivaisesti vaiheesta toiseen ja vaiheiden loputtua tuloksena on valmis sovellus. Evolutionäärisissä malleissa kehitys tapahtuu syklisesti, eli vaiheissa palataan taaksepäin ja tiettyjen vaiheiden välillä kierretään kehää. Jokaisella kierroksella sovellukseen tehdään parannuksia tai tuotetaan uusi ominaisuus. Tarkoituksena on kehittää edellisiä versioita kunnes tuloksena on valmis sovellus. (Pohjonen 2002, 40–41.)

Yleisimmin käytetty vaihejakomalli on vesiputousmalli (kuvio 2), johon kuuluu aina määrittely-, suunnittelu- ja toteutusvaiheet. Usein näiden lisäksi vesiputousmallista löytyy myös esitutkimusvaihe, joka tehdään ennen määrittelyä. Toteutusvaiheen jälkeen testaukselle ja käyttöönotolle sekä ylläpidolle on vielä omat vaiheensa. (Haikala & Märijärvi 2004, 36–37.)



Kuvio 2. Vesiputousmalli.

Lähde: Haikala & Märijärvi 2004, 36

Esitutkimusvaiheessa kartoitetaan asiakasvaatimukset, jotka asettavat yleiset järjestelmätason vaatimukset. Esitutkimuksessa vastataan kysymykseen miksi järjestelmä on toteutettava. Tämä vaihe on ehkä tärkein, sillä asiakasvaatimusten ymmärtäminen sanelee pitkälle oikeanlaisen järjestelmän rakentamisen. Puutteelliset asiakasvaatimukset tai niiden väärinymmärtäminen ei voi tuottaa hyvää ja haluttua lopputulosta. Esitutkimusvaiheessa saadut asiakasvaatimukset analysoidaan ja määrittelyvaiheessa niistä johdetaan suoraan järjestelmävaatimukset, eli otetaan kantaa siihen, millainen ohjelmisto täyttää asiakkaan tarpeet. (Haikala & Märijärvi 2004, 37.)

Määrittelyvaiheessa syntyy toiminnallinen määrittely -dokumentti, jossa kuvataan ohjelmiston sisältämät toiminnalliset ja ei-toiminnalliset vaatimukset ja rajoitteet. Toiminnallisessa määrittelyssä kuvattujen toimintojen toteutus suunnitellaan suunnitteluvaiheessa. (Haikala & Märijärvi 2004, 38, 39.)

Suunnitteluvaihe jaetaan kahteen osaan: Arkkitehtuurisuunnittelussa järjestelmä jaetaan toisistaan riippumattomiin loogisiin kokonaisuuksiin eli moduuleihin. Moduulisuunnittelussa suunnitellaan kunkin moduulin sisäinen rakenne, jossa käsitellään tietomäärittelyitä ja näitä käsitteleviä funktioita. Suunnitteluvaiheen tuotoksena syntyy tekninen määrittely, missä kerrotaan miten ja millä välineillä toiminnalli-

sessä määrittelyssä kuvattu järjestelmä toteutetaan. (Haikala & Märijärvi 2004, 40.)

Toteutusvaiheessa ohjelmisto kirjoitetaan valmiiksi, toimivaksi järjestelmäksi. Testausvaiheessa ohjelmistosta etsitään virheitä yleensä V-mallin mukaisesti. Moduulitestauksessa vikoja etsitään moduuleista, integrointitestauksessa moduulien yhteistoiminnasta ja järjestelmätestauksessa kaikista toiminnoista ja suorituskyvystä. (Haikala & Märijärvi 2004, 40.)

Käyttöönotto ja ylläpito ovat vesiputousmallin mukaan viimeisiä vaiheita. Ylläpitoa on kolmea erilaista. Korjaavassa ylläpidossa korjataan ohjelmavirheitä, adaptiivisessa ylläpidossa huolehditaan järjestelmän sopivuudesta ympäristöönsä eli ohjelmaa muutetaan tarvittaessa ja täydentävässä ylläpidossa ohjelmaan lisätään toimintoja tai muutetaan ohjelmaa sen parantamiseksi. (Haikala & Märijärvi 2004, 41.)

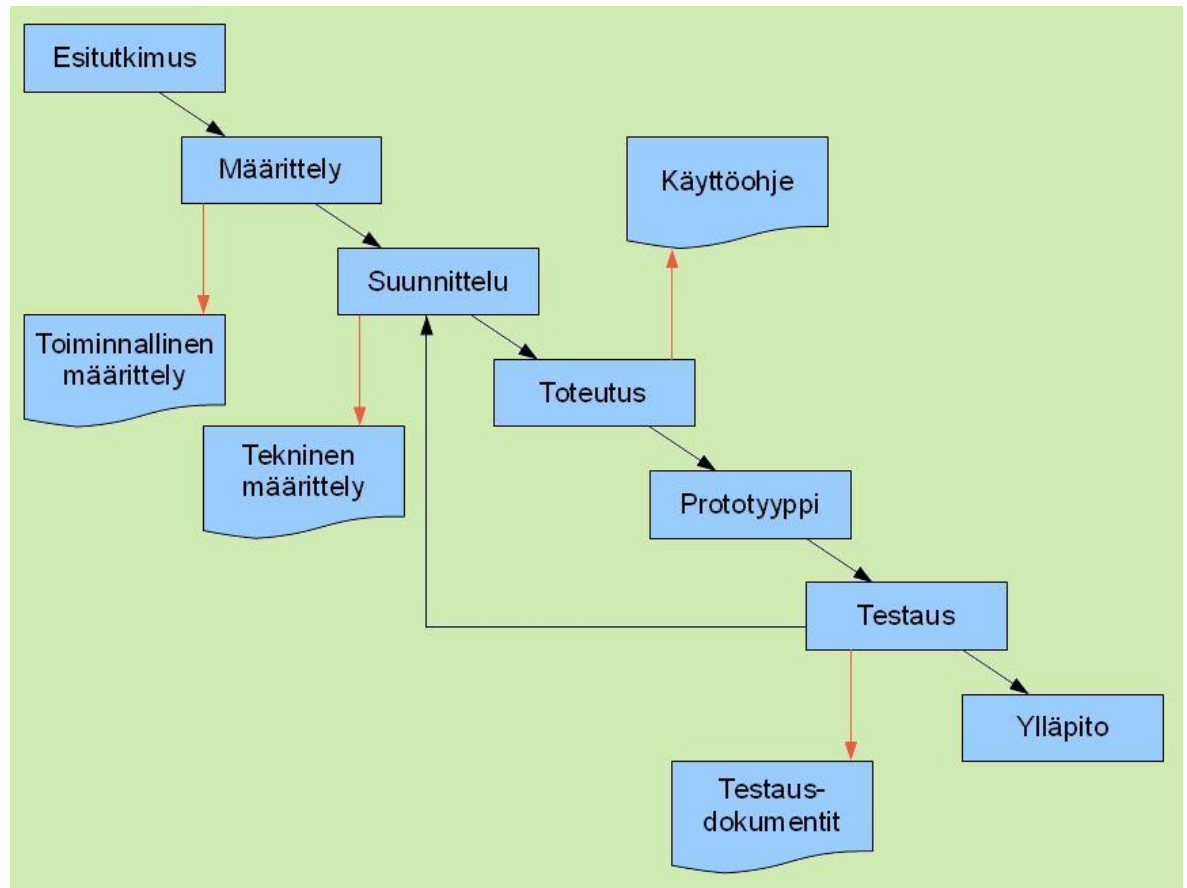
Vesiputousmalli on yleinen ongelmanratkaisumalli, mutta ohjelmistokehitys ei koskaan etene suoraviivaisesti vesiputousmallin mukaisesti. Esimerkiksi asiakasvaatimukset saattavat täydentyä vasta myöhemmin muissa vaiheissa ja usein ne myös muuttuvat. Kuitenkin tämän mallin mukaisesti pyritään toimimaan mahdollisimman pitkälle, sillä sitä voidaan pitää todellisen toiminnan mallina. (Haikala & Märijärvi 2004, 41.)

Tässä työssä ohjelmistokehitystä on lähestytty protoilumallin näkökulmasta. Prototyyppilähestymistapa on sarja vesiputousmalleja. Prototyyppilähestymistavassa kokeillaan jotakin järjestelmän ominaisuutta, ennen kuin varsinainen järjestelmä rakennetaan. Prototyypit ovat hyviä, kun kyseessä on uusi tekninen ratkaisu, ja sen pohjalta tehtävä kokeilu tai kun etsitään epäselviä asiakasvaatimuksia. Prototyypin valmistuttua sen perusteella määritellään toteutettava järjestelmä, joka toteutetaan alusta alkaen uudelleen tai se kehitetään valmiiksi järjestelmäksi. (Haikala & Märijärvi 2004, 42.)

Työssä on edetty protoilumallia (kuvio 3) soveltaen. Rungoksi on otettu vesiputousmallista tuttu rakenne esitutkimuksesta ylläpitoon, ja kustakin vaiheesta on tuotettu niitä vastaavat dokumentit. Toteutusvaiheen tuotoksena on syntynyt soveluksen prototyyppi. Sitä on testattu, ja sen jälkeen on palattu takaisin suunnittelu-



vaiheeseen, jos sovelluksen toimintoihin on kaivattu muutoksia. Mikäli on haluttu lisätä uusia toimintoja, on palattu määrittelyvaiheeseen. Iteraatiota suunnittelusta testaukseen on jatkettu, kunnes on päädytty valmiiseen sovellukseen. Toimimalla kuvan mallin mukaisesti varmistettiin, että asiakas on tyytyväinen käyttöliittymään ja järjestelmässä on kaikki tarvittavat toiminnot.



Kuvio 3. Protoilumalli.

Lähde: Haikala & Märijärvi 2004, 43

Prototyyppilähestymistavan ongelmana on, että asiakas luulee järjestelmän olevan valmis. Prototyyppi saattaa näyttää valmiilta, vaikka suuri osa työstä on vielä kesken. Siksi prototyyppiä ei viimeistellä valmiinnäköiseksi sovellukseksi, kun sitä esitellään asiakkaalle. (Haikala & Märijärvi 2004, 42–43.)

## 2.3 UML

UML (Unified Modeling Language) eli yhtenäistetty mallinnuskieli on Rational Softwaren 1990-luvun loppupuolella kehittämä standardoitu mallinnuskieli. Siinä

käytettävät elementit ja kaaviot perustuvat oliosuuntautuneeseen ajattelumalliin. UML:ää käytetään yleisimmin ohjelmistojen mallintamiseen, mutta käyttökohteita ovat myös mekaanisten laitteiden tai liiketoiminnan kuvaaminen. UML:ssä on yhdeksän erilaista kaaviotyyppiä, joita käytetään järjestelmän kuvaamiseen eri näkökulmista. Järjestelmästä saadaan kokonaiskuva yhdistelemällä kaavioita. (Eriksson & Penker 2000, 4, 6, 7; Larman 2004, 11, 15.)

UML:llä järjestelmävaatimukset voidaan kuvata visualisoiden. Vaatimusten selvittämistä helpottaa käyttötapauskaavion luonti. Vaatimuksia analysoidaan käyttötapausta käyttäen, ja sovelluksen toiminnot kuvataan mahdollisimman yksityiskohteisesti. UML:ää käytetään myös sovelluksen staattisen rakenteen kuvaamiseen. Sovelluksen käsitteet ja niiden väliset yhteydet mallinnetaan luokkakaavioiden avulla, jossa käsitteet heijastavat reaali maailman tietoja. Olioiden väliset vuorovaikutukset ja sovelluksen toiminnot mallinnetaan viestiyhteyksillä eli sekvenssikaavioilla. (Eriksson & Penker 2000, 243, 244.)

### **2.3.1 Käyttötapauskaavio**

Käyttötapauskaaviot (use-case diagram) ovat yksi keskeisimpiä kaavioita UML:ssä, sillä niillä määritellään järjestelmän toimintavaatimukset. Kaikki järjestelmän toiminnot kuvataan käyttötapauskaaviossa, ja tämä tapahtuu aina käyttäjien näkökulmasta, eikä näin ollen ota kantaa siihen, miten toiminnot toteutetaan järjestelmän sisällä. Käyttötapauskaaviossa on kuvattuna toimijat eli käyttäjät sekä itse käyttötapaukset. Käyttötapaukset kuvataan ellipseinä, ja toimijan ja käyttötapausten välistä suhdetta eli assosiaatiota kuvataan viivalla. Yksittäisten käyttötapausten sisältö kuvataan tarkemmin sanallisesti käyttötapauskuvauksissa. Käyttötapauskuvauksesta käy ilmi käyttötapaukseen liittyvä toimija sekä esiehdot, eli millä ehdoilla toiminto voi tapahtua. Käyttötapauskuvauksissa kerrotaan myös kuvaus eli mitä kyseisessä toiminnossa tapahtuu ja frekvenssi eli milloin toiminto tapahtuu. Lisäksi kerrotaan poikkeuksista, mikäli sellaisia liittyy käyttötapaukseen. (Eriksson & Penker 2000, 15, 46, 47.)

### 2.3.2 Sekvenssikaavio

Sekvenssi- eli viestiyhteyksikaavio (sequence diagram) kuvaa yhteistyötä tiettyjen olioiden välillä. Kaavio näyttää sarjan viestejä, eli tarkoituksena on kuvata viestiyhteys olioiden välillä. Yksi sekvenssikaavio kuvaa yhteen käyttötapaukseen liittyvät toiminnot aikajärjestyksessä. Sekvenssikaavio koostuu olioista, jotka ovat kaavion ylälaidassa, sekä niille alaspäin piirretyistä viivoista. Aika kuluu kaaviossa ylhäältä alaspäin. Viestit kuvataan nuolina näiden pystysuorien olioviivojen välille. (Eriksson & Penker 2000, 18.)

### 2.3.3 Luokkakaavio

Luokkakaavioissa (class diagram) on kuvattuna järjestelmän sisältö. Mallin nuselementteinä ovat oliot, luokat ja niiden väliset yhteydet, jotka kaikki ovat tärkeitä oliomallinnuksessa. Olio on asia, jota voi käsitellä ja josta voidaan puhua. Olio on olemassa reaali maailmassa ja se voi olla osa jotakin järjestelmää, kuten organisaatiota. Olio on luokan ilmentymä. Luokalla kuvataan yhden olion ominaisuudet ja käyttäytyminen. Tässä työssä luokkia ovat erilaiset laitteet, esimerkiksi tietokone on yksi luokka. Tietokone-luokan ilmentymiä eli olioita puolestaan ovat tietokantaan tallennettavat yksittäiset tietokoneet. (Eriksson & Penker 2000, 57, 59.)

Luokkakaavio on staattinen, sillä se kuvaa järjestelmän pysyvän rakenteen. Luokat kuvataan kolmeen osaan jaettuna suorakaiteena. Luokkaan sisällytetään sen nimi, attribuutit ja operaatiot. Luokat löytyvät miettimällä, mitä tallennettavia, analysoitavia tai muunnettavia tietoja järjestelmä pitää sisällään. (Eriksson & Penker 2000, 15, 59, 60.)

### 3 TOIMINNALLINEN MÄÄRITETTÄVYYS

Tässä luvussa kuvataan asiakasvaatimukset ja niiden pohjalta on laadittu ohjelmistovaatimukset UML-kaavioita hyödyntäen. Käytettävyyks on järjestelmän ei-toiminnallinen vaatimus.

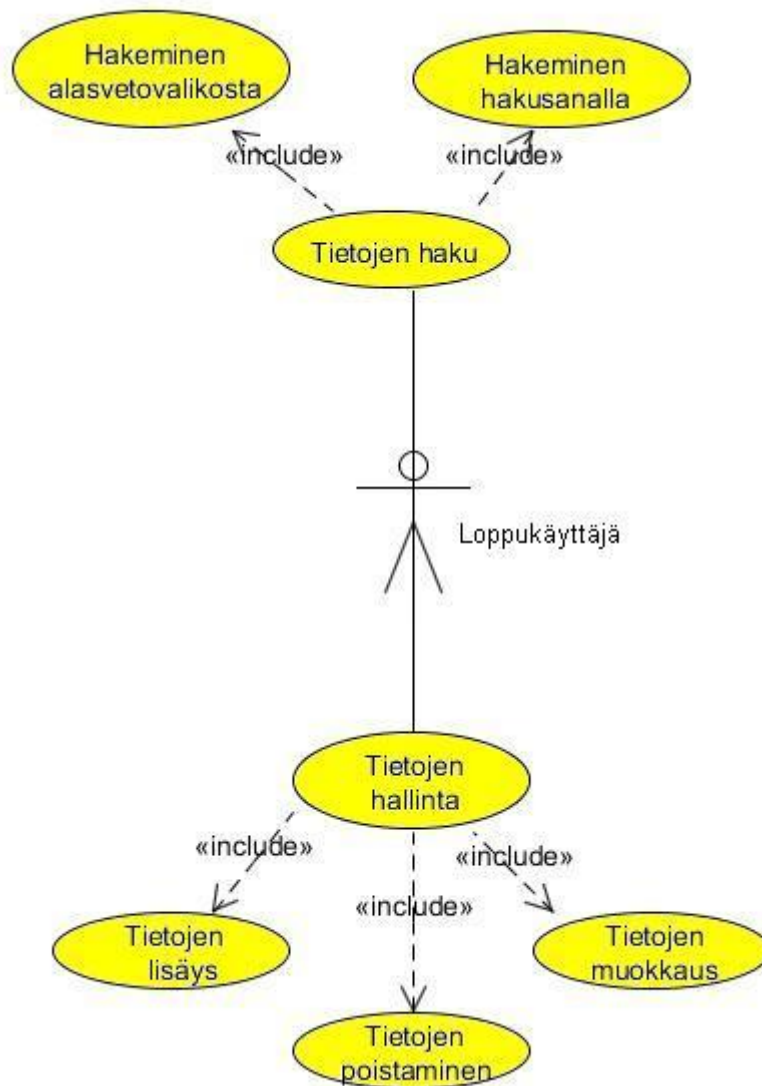
#### 3.1 Asiakasvaatimukset

Asiakasvaatimuksia mietittiin yhdessä loppukäyttäjien eli yksikön atk-tukihenkilöiden, Kimmo Puumalan ja Petri Keskisen kanssa. Tulevalla järjestelmällä tulee pystyä suorittamaan hakutoimenpiteitä, jolloin laitetiedot ovat nopeasti saatavilla. Hakuja suoritetaan käyttäjänimen, ip-osoitteen ja tilanimen perusteella ja tuloksena näytetään tärkeimmät tiedot laitteista ja loput tiedot piilotetaan esimerkiksi painikkeen taakse. Tärkeitä tietoja ovat laitteen nimi/malli, käyttäjän nimi ja tila eli sijainti. Tietokoneesta näytetään lisäksi kovalevyn kapasiteetti, muistin määrä ja suorittimen teho. Tulostintietoihin lisätään kohta, josta näkee kasetin mallin. Tämä nopeuttaa kasetin vaihtoa, kun on mahdollista tarkistaa nopeasti minkä tilan tulostin käyttää mitään kasettia. Lisäksi järjestelmään pystytään lisäämään uusia laitteita ja käyttäjiä, muokkaamaan olemassa olevia tietoja sekä poistamaan tietoja. Tietojen poistamisen tulee tapahtua varmistustoiminnon kautta, jotta tietoja ei voi poistaa vahingossa. Rajoituksena on, että järjestelmän tulee olla Windows-yhteensopiva. Järjestelmässä tulee olla selkeä käyttöliittymä ja ohjelman käytön pitää olla helppoa ja nopeaa.

#### 3.2 Sovelluksen käyttötapaukset

Sovelluksen toimintaympäristö on kuvattu käyttötapauskaaviolla (kuva 4). Sovelluksella on ainoastaan yhdentyypinen toimijaryhmä, loppukäyttäjät. Loppukäyttäjillä tarkoitetaan atk-tukihenkilöitä. Kaikilla loppukäyttäjillä on samat oikeudet sovelluksen kaikkiin toimintoihin, jolloin jakoa pääkäyttäjiiin ja tavallisiin käyttäjiin ei ole tehty. Kaaviosta näkyy, että tietojen hallinta ja haku ovat pääkäyttötapauksia,

ja niiden sisältämät toiminnot on kuvattu "include"-liitoksilla. Käyttötapauskaavio on laadittu sovelluksen käyttäjän näkökulmasta.



Kuvio 4. Sovelluksen käyttötapaukset.

Jokaiselle kaaviossa näkyvälle käyttötapaukselle on laadittu sanallinen kuvaus. Kuvauksessa kerrotaan tarkemmin kuka kyseisen toiminnon suorittaa, missä tilanteessa ja millä ehdoilla se voi tapahtua, toiminnon kuvaus, lopputulos ja poikkeukset. Tässä on esitetty yksi käyttötapauksukuvaus (taulukko 1), loput löytyvät liitteestä 1.

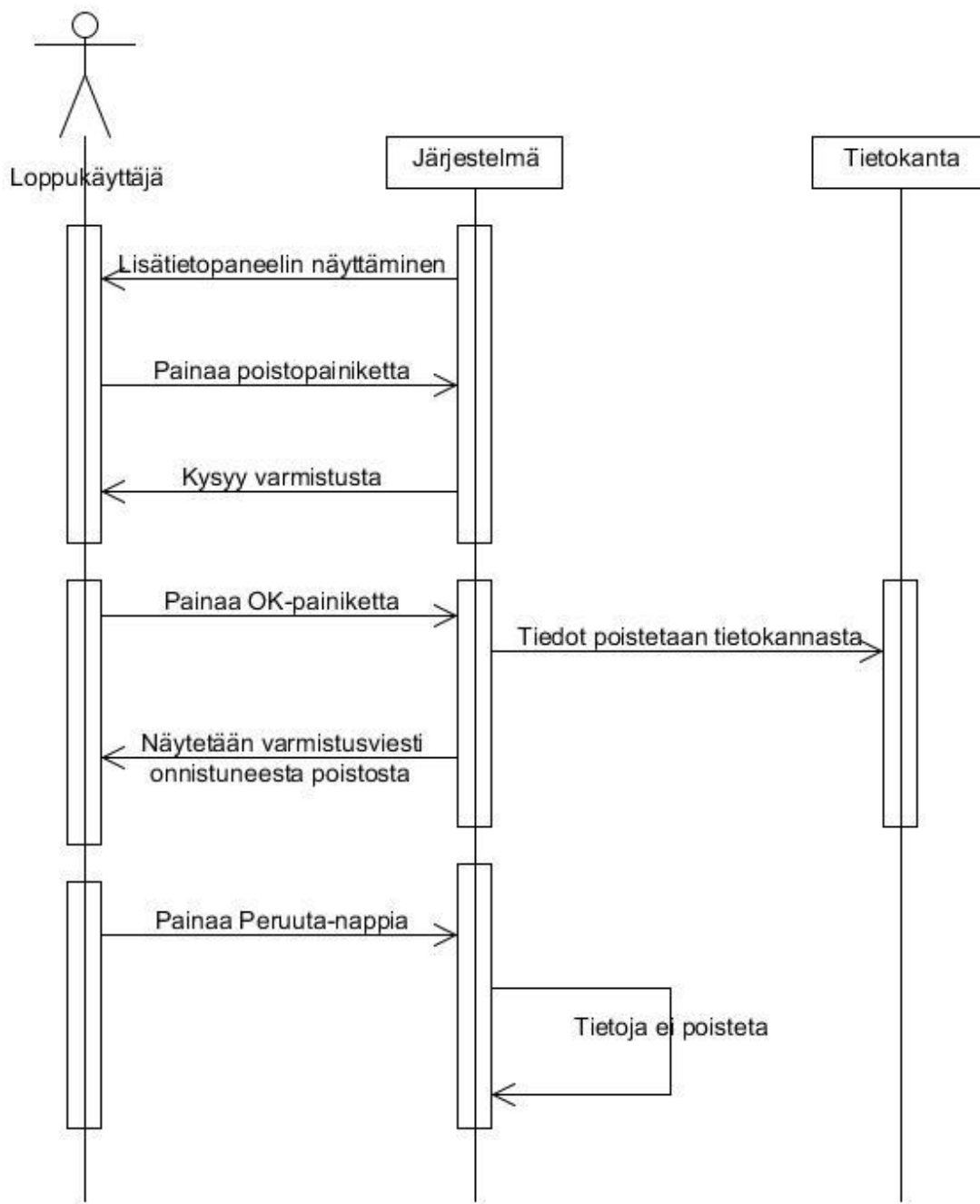
Taulukko 1. Hakeminen hakusanalla -toiminnon käyttötapauskuvauks.

Käyttötapaus: Hakeminen hakusanalla	
Aktori	Loppukäyttäjä
Frekvenssi	Kun halutaan hakea laitetietoja kirjoittamalla hakusana
Esiehdot	Järjestelmä on toimintakuntoinen
Kuvaus	Laitetiedot haetaan kirjoittamalla hakusana ja valitsemalla oikea kategoria alasvetovalikosta (käyttäjänimi, tilanimi, ip-osoite).
Lopputulos	Järjestelmä tulostaa laitetiedot loppukäyttäjälle
Poikkeukset	Mikäli hakusana ja valittu kategoria eivät täsmää, järjestelmä tulostaa tyhjät tulokset käyttäjälle.

### 3.3 Sovelluksen toiminnot

Toiminnot on kuvattu sekvenssikaaviolla. Yksittäisessä kaaviossa kuvataan, mitä tapahtuu tietyssä tilanteessa ja miten aika kuluu tapahtuman sisällä. Tavallisesti kutakin käyttötapauskaavion toimintoa vastaa tietty sekvenssikaavio. Olioiden väliset viestit on esitetty nuolilla, joista selviää viestien suunta. Tavallisimpia viestejä ovat haut, palautukset ja avaamiset. Tässä on esitetty yksi esimerkki sekvenssikaaviosta (kuvio 5) selityksineen, loput löytyvät liitteestä 2.

**Tietojen poistaminen.** Järjestelmä näyttää lisätietopaneelin, jossa on poistopainike joko käyttäjän nimen tai laitteen mallin kohdalla. Loppukäyttäjä painaa poistopainiketta, jolloin järjestelmä pyytää varmistusta loppukäyttäjältä näyttämällä varmistusikkunan. Loppukäyttäjän painaessa OK-painiketta järjestelmä poistaa tiedot tietokannasta. Tämän jälkeen järjestelmä näyttää vahvistuksen tietojen poistosta. Mikäli loppukäyttäjä painaa Peruuta-nappia, järjestelmä lopettaa koodin suorittamisen ja tietoja ei poisteta tietokannasta.



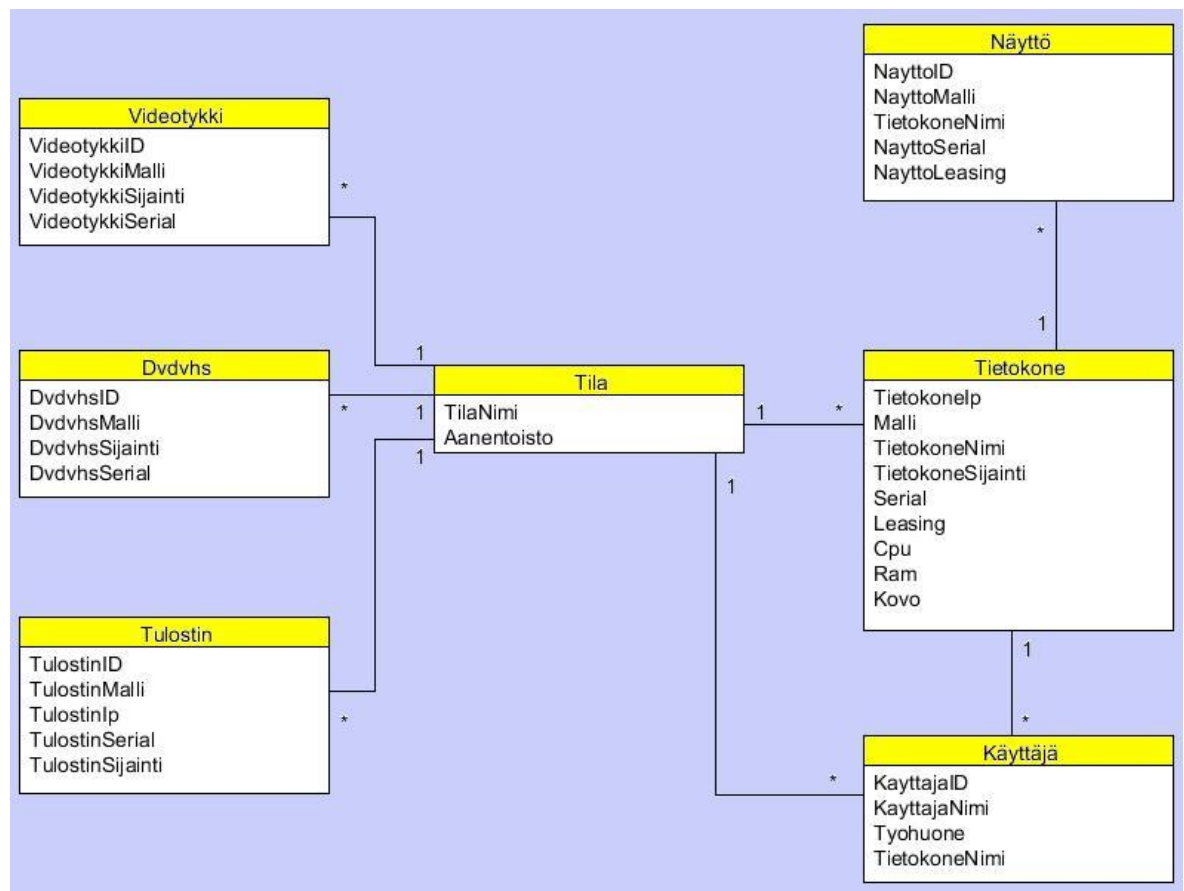
Kuvio 5. Sekvenssikaavio tietojen poistamisesta.

### 3.4 Sovelluksen luokkakaavio

Luokkakaaviota käytetään järjestelmän luokkien etsintään. Tietokantaa kuvaavassa luokkakaaviossa luokat edustavat tietokannan tauluja ja luokan attribuutteina ovat taulun sarakkeet. Taulun tietueet ovat luokan ilmentymiä. Esimerkiksi Tulos-

tin-luokassa attribuutteina voisi olla malli, ip-osoite ja sarjanumero, ja tietokantaan tallennetut yksittäiset tulostimet olisivat Tulostin-luokan ilmentymiä. (Haikala & Märijärvi 2004, 120.)

Kuviossa 6 on asiakasvaatimusten pohjalta hahmotettu luokkakaavio. Suhteet ovat yhden suhde moneen. Luokkiin on lisätty viiteavaimet, jotka saadaan suhteen perusteella. Tietokannan rakennetta kuvaavassa luvussa taulujen välisistä yhteyksistä ja sisällöstä kerrotaan tarkemmin.



Kuvio 6. Sovelluksen luokkakaavio.



## 4 TEKINEN MÄÄRITTELY

Tässä luvussa kerrotaan sovelluksen teknisestä toteutuksesta, web-lomakkeiden toiminnallisuudesta ja sisällöstä sekä tietokantayhteyksistä. Sovellus on tarkoitettu atk-tukihenkilöille päivittäiseksi apuvälineeksi laitetietojen selaamiseen ja muokkaamiseen.

Sovelluksen rakentaminen aloitettiin tietokannan luonnista. Web-sivuston ohjelmointi aloitettiin luomalla MasterPage-sivu, joka muodustui aluksi alavetovalikoista ja painikkeista. Visuaaliseen ilmeeseen ei alkuvaiheessa kiinnitetty suurta huomiota. Sivuston rakentaminen eteni sivu kerrallaan. Yhden sivun valmistuttua se pyrittiin tekemään mahdollisimman valmiiksi ennen siirtymistä uuden sivun rakentamiseen. Uusien laitteiden ja käyttäjien lisäyssivut olivat melko samanlaisia, joten ensimmäisenä rakennettua lisäyssivua pystyi hyödyntämään säästäten aikaa ja vaivaa. Lopuksi käyttöliittymää paranneltiin ja sivujen ulkoasu viimeisteltiin.

### 4.1 Sovelluksen yleiskuvaus

Web-sovellus on rakennettu Visual Studio 2008 -sovelluskehittimellä ja ohjelmointikielenä on käytetty C#-kieltä. Sovelluksen pohjalla toimiva tietokanta on luotu Microsoft Access -tietokantatyökalulla. Tietokantayhteydet ja tietokantaoperaatiot on toteutettu sovellukseen käyttäen Microsoftin ADO.NET (ActiveX Data Objects) -tekniikkaa.

### 4.2 Relaatiotietokannat

Tietokanta on kokoelma yhteenliittyvää tietoa. Tietokantoja voi olla pieniä ja yksinkertaisia tai suuria ja monimutkaisia. Tietokannat pohjaavat aina johonkin reaali maailman tapahtumaan, joista tietokantaan saadaan tietosisältö. Tiedot tallennetaan tietokantaan jotain tiettyä tarkoitusta ja käyttäjiä varten. Reaalimaailmassa tapahtuvat muutokset heijastuvat tietokantaan. (Lahtonen 2004, 2.)

Relaatiomalli jaetaan kolmeen osaan: rakenne, käsittely ja eheyssäännöt.

#### 4.2.1 Rakenne

Relaatiotietokantojen peruselementti on taulu, jota kutsutaan myös taulukoksi, johon tallennetaan tietty asiakokonaisuus, esimerkiksi Tietokone-tiluun tallennetaan kaikki tietokonetta koskevat tiedot. Taulut koostuvat kentistä eli sarakkeista ja riveistä eli tietueista. Jokaiselle kentälle määritetään tietotyyppi, esimerkiksi numeerinen tai merkkijono, ja pituus. Jokaisella kentällä on yksilöivä nimi, ja sen sisältämällä tiedolla on yhteinen tietotyyppi eli ne kuuluvat samaan arvojoukkoon. Jokaisessa taulussa on oltava tunnistekenttä eli perusavain, joka yksilöi taulun sisältämät tietueet. Joihinkin kenttiin tallennettavat tiedot voivat olla riittävän yksilöiviä, jolloin kenttä käy perusavaimesta. Tällöin täytyy olla varma, ettei samannimistä tietoa voi esiintyä kahdesti kentässä. Tästä on esimerkkinä Tietokone-tilu, jossa TietokoneNimi-kenttä yksilöi jokaisen sinne tallennetun tietokoneen. Jokaisella tietokoneella on siis yksilöivä nimi. Mikäli taulusta ei löydy sopivaa perusavainkenttää, sellainen on luotava. Esimerkiksi Naytto- ja Tulostin-tiluissa perusavainkenttä on pitänyt erikseen luoda, koska taulusta ei löydy valmiiksi yksilöivää kenttää. Kaksi taulua yhdistetään toisiinsa viiteavaimen avulla. Tällöin puhutaan lapsi- ja äititilusta. Lapsitaulun viiteavaimeksi asetetaan äititaulun perusavain. Se taulu, mistä viitataan, on lapsitaulu, ja viittauksen kohteena oleva taulu on äititaulu (Hovi 2004, 6, 8; Hovi & Huotari 2005, 8,-10).

#### 4.2.2 Käsittely

Käsittelyllä tarkoitetaan tietokannan tietoihin kohdistuvia hakuja ja päivityksiä. Käsittelyssä tulee esiin joukko-opillisuus, mikä tarkoittaa, että taulun muodostuessa joukosta rivejä, kyseiseen joukkoon voi kohdistaa joukko-operaatioita, esimerkiksi hakuja. Joukko-operaatiolla voi käsitellä yhden tai useamman taulun. Käsittely toteutetaan relaatiotietokannoissa SQL-kyselykielellä, joka nojautuu joukko-oppiin. (Hovi 2004, 8.)

SQL (Structured Query Language) syntyi IBM:n laboratoriossa San Josessa 70-luvulla, ja siitä on tullut laajimmin relaatiotietokantojen yhteydessä käytetty kieli. SQL standardoitiin ensimmäisen kerran vuonna 1986 ja siihen tehtiin parannuksia myöhemmin. SQL-kieltä käytetään tietokannan rakenteen määrittelyssä ja muut-

tamisessa, kyselyiden tekemisessä, päivityskomennoissa eli tietojen lisäämisessä, poistamisessa ja muuttamisessa, turvallisuuden ja valtuuksien hoidossa, upotetun SQL:n hallitsemiseen ja tapahtumankäsittelyn ohjaamiseen. Näistä kyselyt ovat yleisin käyttöalue. DDL:sta (Data Definition Language) puhuttaessa tarkoitetaan SQL:n osaa, joka liittyy tietokannan rakenteen määrittelyyn ja muuttamiseen. DML:sta (Data Manipulation Language) puhuttaessa taas tarkoitetaan SQL:n osaa, joka liittyy kyselyiden ja päivityksien tekoon. SQL:ää voi käyttää mm. vuorovaikutteisesti, upotetusti tai dynaamisesti. Työssä käytettiin kahta ensimmäistä käyttötapaa. Vuorovaikutteista SQL:ää (interactive SQL) käytetään mm. SQL:n testaamiseen ja opetteluun. Kun sovelluksen Access-tietokanta oli saatu valmiiksi, testattiin sen toimivuutta muutamilla sql-kyselyillä. Ennen kuin tarvittavat sql-kyselyt liitettiin ohjelmakoodiin, tarkistettiin niiden antaman tuloksen oikeellisuus suorittamalla kyseiset kyselyt. Upotetussa SQL:ssä (embedded SQL) SQL-lauseet upotetaan ohjelmakoodin sekaan ja kyselyn antamat tulokset tallennetaan ohjelmointikielen muuttujiin. (Lahtonen 2004, 38, 39; Hovi & Huotari 2005, 10.)

#### **4.2.3 Eheyssäännöt**

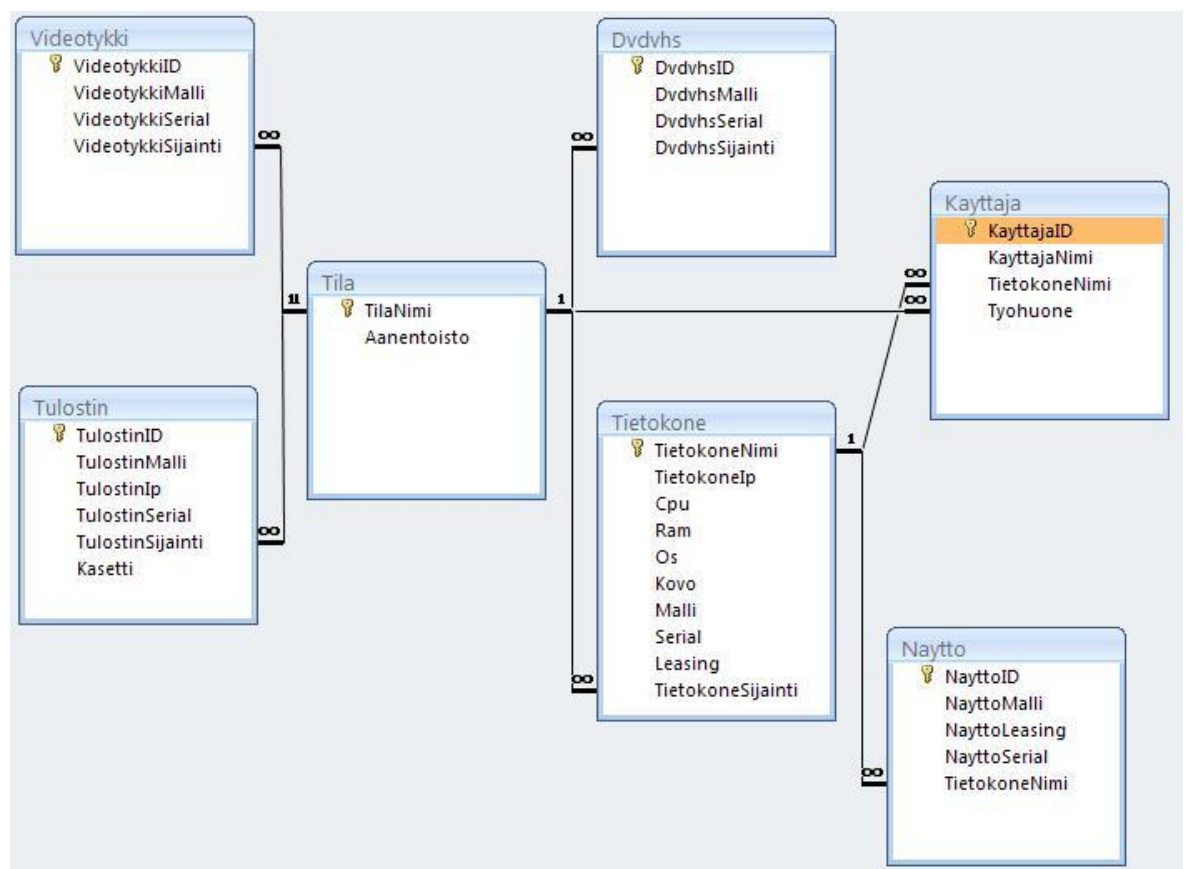
Eheyssäännöillä tarkoitetaan Coddin määrittelemiä avain- ja viite-eheyttä. Avain-eheydellä tarkoitetaan, että perusavaimen arvo on pakollinen. Perusavaimella ei voi olla NULL-arvoja. Viite-eheydellä tarkoitetaan, että taulujen väliset yhteydet on luotu siten, ettei viite-eheys säry. Esimerkiksi jos Tila- ja Käyttäjä-taulun välillä ei olisi vyörytettyä poistoa, tilaa poistettaessa Käyttäjä-tauluun tallennetut käyttäjät jäisivät ”orvoiksi” ja viite-eheys rikkoutuisi. Viite-eheyden yhteydessä puhutaan lapsi- ja äititauluista. Viittaavasta taulusta eli lapsitaulusta täytyy löytyä viiteavainkenttä, joka on viittauksen kohteena olevan taulun eli äititaulun perusavain. (Hovi 2004, 9,10; Hovi & Huotari 2005, 11, 12.)

#### **4.2.4 Tietokannan rakenne**

Sovelluksen tietokanta on toteutettu Microsoft Access 2007 -tietokantatyökalulla. Tietokannan eheys on varmistettu viiteavaimia sekä johdannaispäivityksiä ja -

poistoja käyttämällä. Lisäksi perusavainkentissä ei sallita kaksoisarvoja. Uutta tietoa lisättäessä ohjelma tarkistaa pakolliset kentät, jolloin ongelmien syntyminen minimoidaan. Tietyt tekstikentät, joita vastaavat tietokannan kentät on määritelty pakollisiksi tiedoiksi, validoidaan siten, että niitä ei voi jättää tyhjiksi, tai muuten ohjelma ei suorita lisäystä. Tietokannan taulujen kentille ei sallita NULL-arvoja. Tietyille kentille on asetettu oletusarvo sovelluksen käyttöä helpottamaan. Esimerkiksi uutta tietokonetta lisättäessä loppukäyttäjä syöttää itse tärkeimmät tiedot, eli tietokoneen nimen, tilan, käyttäjän nimen ja ip-osoitteen. Muut tiedot, esimerkiksi tietokoneen malli ja sarjanumero eivät ole siinä määrin tärkeitä tietoja, että loppukäyttäjä haluaisi tallentaa ne saman tien tietokantaan. Tällaisissa tilanteissa oletusarvot ovat erittäin käyttökelpoisia, koska ne tekevät tietyltä osin sovelluksen käytön joustavaksi. Oletusarvojen ansiosta loppukäyttäjän ei tarvitse syöttää kaikkia tietoja yhdellä kertaa, eikä tästä synny ongelmia tallennettaessa tietoja tietokantaan.

Kuviossa 7 on havainnollistettu tietokannan taulujen väliset yhteydet.



Kuvio 7. Tietokannan yhteydet.

Tila-tiluun tallennetaan kaikkien tilojen nimet sekä opetustilan ollessa kyseessä äänentoistomahdollisuus. Tila-tilun perusavain on TilaNimi, joka yksilöi kunkin tilan.

Tietokone-tilu on Tila-tilun lapsitilu. Tietokone-tilu pitää sisällään kaikki tietokoneiden tiedot. Perusavaimena on TietokoneNimi, joka on tekstityyppiä. Tietokone-tilulle ei ollut tarvetta määritellä perusavaimeksi juoksevaa laskurikenttää, sillä tietokoneiden nimet riittävät yksilöimään kunkin tietokoneen. Viiteavaimena on TietokoneSijainti, joka viittaa tiettyyn tilaan Tila-tilussa. Yhteystyyppi on yhden-suhde-moneen.

Dvdvhs-tilu sisältää dvd-soittimien ja vhs-laitteiden tiedot. Dvdvhs-tilu on Tila-tilun lapsitilu. Perusavaimena on DvdvhsID, joka yksilöi kunkin laitteen tilusta. Viiteavain DvdvhsSijainti viittaa tiettyyn tilaan Tila-tilussa. Yhteystyyppi on yhden-suhde-moneen.

Videotykki-tilu on Tila-tilun lapsitilu, jonne tallennetaan videotykkien tiedot. Perusavain VideotykkiID yksilöi kunkin videotykin ja siihen liittyvät tiedot. Viiteavain VideotykkiSijainti viittaa tiettyyn tilaan Tila-tilussa. Yhteystyyppi on yhden-suhde-moneen.

Tulostin tilu on Tila-tilun lapsitilu. Tulostin-tiluun tallennetaan tulostimien tiedot. Perusavaimena on TulostinID, joka yksilöi kaikki tulostimet. Viiteavain TulostinSijainti viittaa Tila-tiluun. Kukin tulostin liittyy tiettyyn tilaan. Yhteystyyppi on yhden-suhde-moneen.

Näyttö-tiluun tallennetaan näyttölaitteiden tiedot. Perusavaimena on Nayttoid, laskurityyppinen kenttä, joka yksilöi tilun tietueet. Viiteavaimena on TietokoneNimi, joka viittaa tiettyyn Tietokone-tilun tietokoneeseen. Yhteystyyppi on yhden-suhde-moneen. Näyttö-tilu on Tietokone-tilun lapsitilu.

Käyttäjä-tiluun tallennetaan kaikkien käyttäjien tiedot. Käyttäjä-tilu on Tila-tilun lapsitilu. Perusavain KayttajaID yksilöi Käyttäjä-tilun tietueet. Viiteavain Tyohuone viittaa tiettyyn tilaan Tila-tilussa. Tilun toinen viiteavain TietokoneNimi viittaa tiettyyn tietokoneeseen Tietokone-tilussa. Molemmissa yhteyksissä

yhteystyyppi on yhden-suhde-moneen. Kukin käyttäjä työskentelee yhdessä tilassa, mutta yhdessä huoneessa voi työskennellä useampi henkilö.

### 4.3 Sovelluksen rakenne

Sovellus on rakennettu ASP.NET (Active Server Pages) -tekniikalla. Täten sovellus on dynaaminen web-sivusto, joka sisältää toimintoja, joita käyttäjä voi suorittaa. Sivusto koostuu etusivusta, useista web-lomakkeista ja MasterPage-sivusta. MasterPage-sivulla määritetään sivuston visuaalinen ilme. Kaikki web-lomakkeet käyttävät samaa MasterPage-sivua, joten sivun visuaalinen ilme pysyy koko ajan samana. Näin sivusto saatiin selkeäksi ja loppukäyttäjän ei tarvitse navigoida sivulta eli web-lomakkeelta toiselle

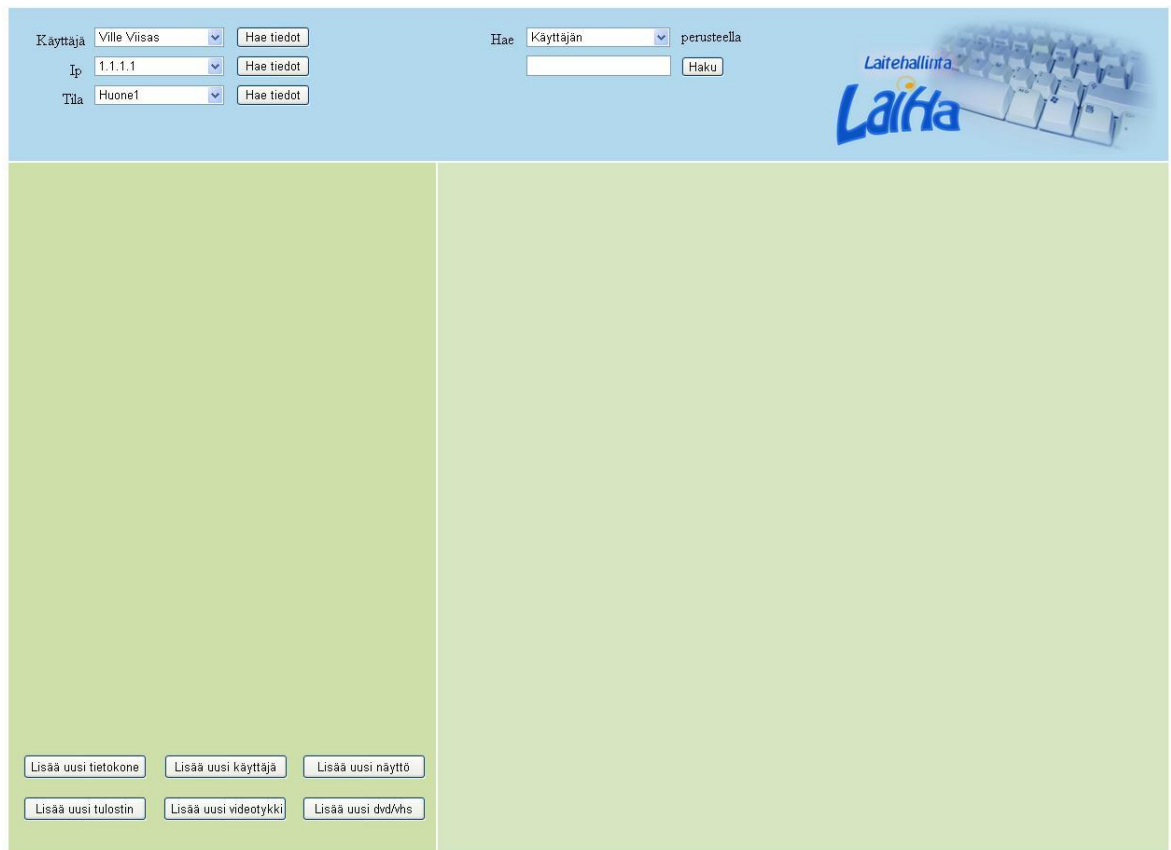
#### 4.3.1 Käyttöliittymä

Sovelluksessa on käytetty MasterPage-tekniikkaa käyttöliittymän luomisessa. Kun sovellus käynnistetään, avautuu etusivu, jonka ylälaudassa näkyy sovelluksen logo ja hakutoiminnot sekä alalaidassa laitteiden ja käyttäjän lisäyspainikkeet, jotka kaikki ovat MasterPage-sivulle lisättyjä komponentteja. Lisäksi etusivulla näkyy teksti "Tervetuloa! Tämä on etusivu!". Logo toimii linkkinä etusivulle. Sovelluksessa on käytetty useampaa web-lomaketta eli sivua, joiden pohjana kyseinen MasterPage on. Jokaisen sivun ylälaudassa näkyy aina logo ja hakutoiminnot. MasterPage-sivulla on yksi ContentPlaceHolder-kontrolli (kuvio 8), joka määrittää yksittäisen sivun paikan ja sisällön MasterPage-sivulla. Jokaisella web-lomakkeella on content-tagit, joiden sisään lomakkeiden sisältö määritellään. Tägeissa on viittaus MasterPage-sivuun. Kun web-lomake avataan, ASP.NET yhdistää sen MasterPage-sivuun. Kaikki web-lomakkeet avautuvat samaan ContentPlaceHolder-kontrolliin, joten loppukäyttäjällä voi olla näkyvissä vain yksi sivu kerrallaan.



Kuvio 8. MasterPage-sivun rautalankamalli.

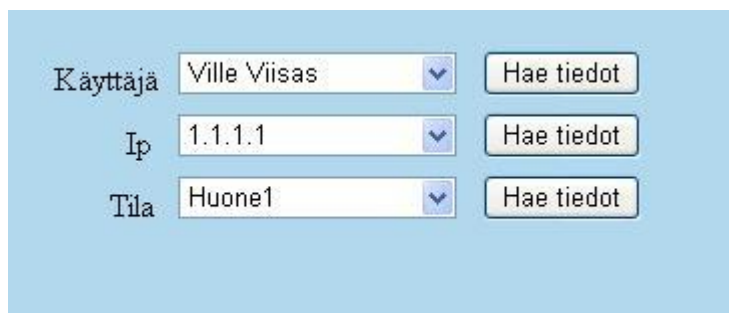
Etusivuna (kuvio 9) toimiva Default.aspx-sivu näytetään loppukäyttäjälle aina ensimmäisenä ja siinä näkyy MasterPage-sivun sisältö. Alasvetovalikkohaussa järjestelmä lataa käyttäjien nimet, ip-osoitteet ja tilojen nimet niitä vastaaviin alasvetovalikoihin.



Kuvio 9. Sivuston etusivu.

#### 4.3.2 Haku alasvetovalikoista

Laitetietoja voi hakea alasvetovalikkoja hyödyntäen. (kuvio 10). Haun voi suorittaa käyttäjän nimen, ip-osoitteen tai tilanimen perusteella. Tässä on käytetty esimerkkinä käyttäjänimen perusteella tapahtuvaa hakua.



Kuvio 10. Alasvetovalikot tietojen hakemista varten.

Kun sovellus avataan, ohjelma hakee tietokannasta kaikkien käyttäjien (Esimerkki 1) ja tilojen nimet ja ip-osoitteet. Haku suoritetaan MasterPage-sivulla ja sitä var-



ten on luotu refresh-niminen metodi, jota kutsutaan PageLoad-osiossa. Alasvetovalikoiden täyttämistä ei voitu suorittaa PageLoad-osiossa, sillä muutoin uudet lisätyt tiedot eivät olisi päivittyneet alasvetovalikoihin. Haetut tiedot asetetaan muuttujiin, joilla puolestaan täytetään käyttäjä-, ip- ja tila-alasvetovalikot. Loppukäyttäjän saapuessa etusivulle alasvetovalikot on täytetty valmiiksi tiedoilla. Loppukäyttäjä voi suorittaa yhden haun kerrallaan valitsemalla joko käyttäjänimen, ip-osoitteen tai tilanimen sitä vastaavasta alasvetovalikosta.

#### Esimerkki 1. Alasvetovalikon täyttäminen käyttäjänimillä.

```
//tällä lauseella haetaan käyttäjien nimet käyttäjä-alasvetovalikkoon
string sql = "Select KayttajaNimi from Kayttaja";

//luodaan tietokantayhteys
using (OleDbConnection connection = new OleDbConnection
    ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=G:\\ont\\Ohjelma\\LaiHa\\Laitteet5.mdb"))
{
    //kerrotaan mikä sql-lause suoritetaan ja mitä tietokantayhteyttä käytetään
    OleDbCommand command = new OleDbCommand(sql, connection);
    //avataan tietokantayhteys
    connection.Open();

    //luodaan lukija, joka käy lukemassa tiedot tietokannasta
    OleDbDataReader reader = command.ExecuteReader();

    while (reader.Read())
    {
        //luetaan tiedot tietokannasta ja lisätään tiedot muuttujaan
        string nimi = (string)reader["KayttajaNimi"];
        //lisätään muuttujan sisältämä tieto alasvetovalikkoon
        DropKayttaja.Items.Add(nimi);
    }
    reader.Close();
    //suljetaan tietokantayhteys
    connection.Close();
}
```

Käyttäjällä voi olla useampia samannimisiä näyttöjä. Näytön lisätietojen hakeminen toteutettiin hakemalla niiden id:t piilokenttään. Tätä varten luotiin uusi item-niminen ListItem-olio (Esimerkki 2), jossa olion text-ominaisuudeksi haettiin näytön malli, ja value-ominaisuudeksi näytön id. Sen jälkeen näyttöalasvetovalikko täytetään item-oliolla, joka sisältää sekä loppukäyttäjälle näkyvän tekstin että loppukäyttäjältä piilossa olevan tunnisteen. Ensimmäisenä alasvetovalikossa olevan näytön id sijoitetaan tunnistetekstikenttään alasvetovalikon SelectedItem.Value-ominaisuuden avulla.

Esimerkki 2. Uuden ListItem-olion luonti.

Lähde: Microsoft Corporation, [Viitattu 20.1.2011].

```
ListItem item = new ListItem();
item.Text = reader["NayttoMalli"].ToString();
item.Value = reader["NayttoID"].ToString();
DropNaytto.Items.Add(item);
```

### 4.3.3 Haku hakusanalla

Sovelluksessa on toinenkin tapa hakea tietoja. Loppukäyttäjä voi hakea tietoja myös kirjoittamalla hakusanan (kuvio 11) ja valitsemalla haettavan sanan kategorian (käyttäjä, ip-osoite tai tila). Kategoria valitaan hakusanakentän yläpuolella olevasta alasvetovalikosta, johon on asetettu oletuksena äsken mainitut vaihtoehdot. Vaihtoehtojen value-ominaisuudeksi asetetaan täsmälleen tietokannan saraketta vastaava arvo. Tämän perusteella ohjelmalle kerrotaan, mistä sarakkeesta tiedot haetaan. Esimerkiksi "Käyttäjän"-vaihtoehdon value-ominaisuudeksi on asetettu KayttajaNimi, joka vastaa Käyttäjä-taulun KayttajaNimi-saraketta. Hakeminen on toteutettu piilokentän avulla. Loppukäyttäjän syöttämä arvo haetaan tietokannasta ja se tulostetaan piilokenttään. Tällä tarkistetaan, onko syötettyä arvoa olemassa, ja voidaanko näin ollen hakea sen perusteella laitetietoja tietokannasta. Jos piilokenttään tulostuu arvo, se välitetään hakutulossivulle.



Kuvio 11. Hakusanahaun lomakekomponentit.

Ohjelmaan on rakennettu valittu-muuttuja, jonka arvoksi asetetaan kategoriasalasvetovalikosta valittu vaihtoehto. Valittu kategoria tarkistetaan if-lauseilla (Esimerkki 3), ja haku suoritetaan sen perusteella TbTulos-piilokenttään, jonka arvo puolestaan välitetään hakutulossivulle.

### Esimerkki 3. Valitun kategorian tarkistaminen if-lauseella.

```
//valittu-muuttuja saa arvokseen alasvetovalikosta valitun arvon
string valittu = DropValinta.SelectedValue;
//jos valitun arvo on "KayttajaNimi", suoritetaan if-lauseen sisällä oleva koodi
if (valittu == "KayttajaNimi")
{
    //hakusana-muuttuja saa arvokseen TbHakusana-tekstikenttään syötetyn tekstin
    string hakusana = TbHakusana.Text;

    //lauseessa "valittu"-sanon tilalla on tässä tapauksessa sana "KayttajaNimi",
    //eli haetaan käyttäjänimen perusteella
    string sql = "Select KayttajaNimi from Kayttaja where " + valittu + " = '" + hakusana + "'";

    using (OleDbConnection connection = new OleDbConnection
        ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=G:\ont\Ohjelma\LaiHa\Laitteet5.mdb"))
    {
        OleDbCommand command = new OleDbCommand(sql, connection);
        connection.Open();

        OleDbDataReader reader = command.ExecuteReader();

        while (reader.Read())
        {
            //luetaan käyttäjänimi tietokannasta kayttajaNimi-muuttujaan
            string kayttajaNimi = (string)reader["KayttajaNimi"];
            //haettu käyttäjänimi tulostetaan näkymättömään tekstikenttään
            TbTulos.Text = kayttajaNimi;
        }
        reader.Close();
        connection.Close();
    }

    //välitetään tekstikentän arvo hakutulossivulle
    Response.Redirect("Kayttaja_hakutulos.aspx?TbTulos=" + TbTulos.Text);
}
```

Hae tiedot -painikkeella loppukäyttäjä ohjataan hakutulossivulle (kuvio 12) Response.Redirect-metodia käyttäen. Metodissa määritellään ensin sivu, jota kutsutaan ja sen perään on koottu lähetettävien lomakekontrollien arvot. Esimerkissä lähetettävä arvo on piilokentästä otettu käyttäjänimi.



The screenshot shows a search results page with a light green background. It contains several input fields and buttons arranged in a grid-like fashion. The fields are labeled on the left and contain text or a dropdown arrow. To the right of each main field is a button labeled 'lisätiedot' (more info).

Field Label	Field Value	Action Button
Käyttäjä	Ville Viisas	Käyttäjän lisätiedot
Työhuone	Huone2	
Tietokone	Villenkone	Tietokoneen lisätiedot
Cpu	dwe	
Ram	2 Gb	
Kovalevy	Maxtor	
Tulostin	Brother	Tulostimen lisätiedot
Tulostimen ip	2.3.4.5	
Näyttö	Benq	Näytön lisätiedot

Kuvio 12. Hakutulossivu, kun haku on tehty käyttäjänimen perusteella.

Hakutulossivulla valittu arvo otetaan vastaan PageLoad-osiossa Request-oliolla ja se asetetaan suoraan kayttaja-muuttuun. Koska hakutapoja on kaksi, oli tarpeen luoda if-lause (Esimerkki 4), jolla tarkistetaan, kumpaa hakutapaa on käytetty. Näin jouduttiin tekemään, sillä molemmista hakutavoista päädytään samalle hakutulossivulle. Yhdellä sql-lauseella haetaan kaikki valittuun käyttäjään liittyvät laite-tiedot ja KayttajaNimi saa arvonsa kayttaja-muuttujasta.

Esimerkki 4. If-lauseella tarkistetaan, kumpaa hakutapaa on käytetty.

```
//jos vastaanotettu tieto on saatu tekstikentän, eli
//hakusanahaun kautta, suoritetaan if-lauseen sisällä oleva koodi
if (Request["TbTulos"] != null)
{
    string sql2 = "SELECT distinct KayttajaID, KayttajaNimi, Tyohuone, Kayttaja.TietokoneNimi, Cpu,"
        + "Ram, Kovo, TulostinID, TulostinMalli, TulostinIp, NayttoMalli, NayttoID "
        + "from Kayttaja, Tietokone, Tulostin, Naytto "
        + "where Naytto.TietokoneNimi = Kayttaja.TietokoneNimi "
        + "and TulostinSijainti = Tyohuone "
        + "and Tietokone.TietokoneNimi = Kayttaja.TietokoneNimi "
        + "and KayttajaNimi='" + kayttaja2.ToString() + "'";
```

#### 4.3.4 Poista käyttäjä/käyttäjän lisätiedot

Hakutulossivujen haluttiin sisältävän vain tärkeimmät tiedot. Tätä varten sovellukseen lisättiin Panel-komponentti, jonka sisään laitettiin tekstikentät, joihin tulostuu kaikki laitteen/käyttäjän muut tiedot, myös ne jotka näytetään ensimmäisellä kerralla hakutulossivun avautuessa. Lisätietopaneeli on asetettu oletusarvoisesti näkymättömäksi, ja se tulee näkyviin vasta kun painaa jonkin laitteen/käyttäjän lisätietopainiketta. Käyttäjän lisätiedot -painikkeen vieressä on TbKayttajaID-piilokenttä, johon haetaan käyttäjän id eli tunniste tietokannasta. Tämän avulla ohjelma yksilöi samannimiset käyttäjät. Järjestelmä osaa näin hakea tietokannasta oikean käyttäjän tiedot ja näyttää ne lisätietopaneelissa.

Käyttäjä poistetaan lisätietojen tarkastelun yhteydessä. Käyttäjiä voi poistaa ainoastaan yhden kerrallaan. Poista käyttäjä -painikkeessa on OnClientClick-ominaisuus, jossa kutsutaan JavaScript-kielen confirmAction()-funktiota (Esimerkki 5). Funktiolla muodostetaan varmistusikkuna, joka näytetään ennen varsinaista poistamista. Ikkunanluontikoodi (Esimerkki 6) on laitettu sivun Content-osion alkuun ja poiketen muusta koodista sitä ei suoriteta palvelimella vaan selaimella.



### Esimerkki 5. Painikkeen OnClientClick-ominaisuus.

```
<asp:Button ID="BtnPoistaKayttaja" runat="server"
    onclick="BtnPoistaKayttaja_Click"
    OnClientClick="javascript:return confirmAction();"
    style="top: 5px; left: 286px; position: absolute; height: 25px; width: 100px"
    Text="Poista käyttäjä" />
<asp:TextBox ID="TbLisaRam" runat="server"
```

### Esimerkki 6. Varmistusikkunan luonti JavaScriptin confirm-funktiolla.

Lähde: Javascript confirm, [Viitattu 22.1.2011].

```
<%--Javascriptin innerhtml-ominaisuudella muutetaan LblVahvistus:n tekstiä.
    Confirm metodille annetaan paluuarvo eli viesti, mikä tulostetaan jos Ok-nappia on painettu.
    Ok-napin painamisen jälkeen koodin suoritus jatkuu--%>
<script type="text/javascript">

    function confirmAction()
    {
        //kutsutaan confirm-funktiota varustettuna hipsukoiden sisällä olevalla viestillä
        if (confirm('Haluatko varmasti poistaa tiedot? Poista tiedot painamalla OK.'))
        {
            //OK-nappia on painettu
            return true;
        }
        else
        {
            //Peruuta-nappia on painettu, vahvistusviestiä ei näytetä
            document.getElementById('<%=LblVahvistus.ClientID %>').innerHTML = "";
            return false;
        }
    }

</script>
```

Varmistusikkunassa (kuvio 13) on OK- ja Peruuta-painikkeet. Mikäli painetaan OK-painiketta, koodin suoritus jatkuu ja tiedot poistetaan delete-lauseella (Esimerkki 7). Käyttäjätietoja poistettaessa poistuu tietokannasta myös käyttäjän tietokoneen tiedot. Tietokantaan ei näin ollen jää tietokoneita, joilta puuttuu käyttäjä. Käyttäjän tietokoneen poistaminen tapahtuu erillisellä delete-lauseella. Tietokone poistetaan nimen perusteella, joka saadaan TbTietokone-tekstikentästä. Poistamisen jälkeen tulostetaan varmistusviesti LblVahvistus-muuttujaan.



Kuvio 13. Varmistusikkuna.

Esimerkki 7. Käyttäjän ja käyttäjälle kuuluvan tietokoneen poistaminen delete-lauseella.

```
//käyttäjä poistetaan käyttäjänimen perusteella
string sql = @"delete from Kayttaja where KayttajaNimi='" + TbKayttaja.Text + "'";

using (OleDbConnection connection = new OleDbConnection
    ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=G:\\ont\\Ohjelma\\LaiHa\\Laitteet5.mdb"))
{
    OleDbCommand command = new OleDbCommand(sql, connection);
    connection.Open();
    command.ExecuteNonQuery();
    connection.Close();
}

//tällä lauseella poistetaan käyttäjälle kuuluva tietokone
string sql2 = @"delete from Tietokone where TietokoneNimi='" + TbTietokone.Text + "'";

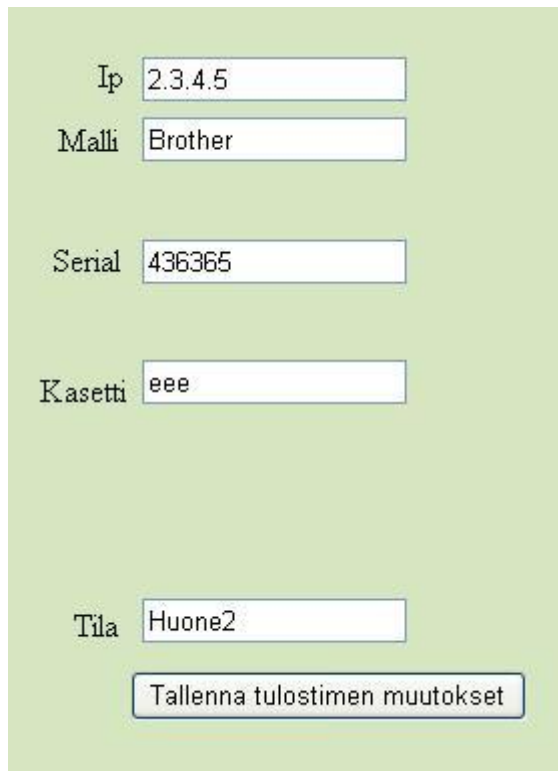
using (OleDbConnection connection = new OleDbConnection
    ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=G:\\ont\\Ohjelma\\LaiHa\\Laitteet5.mdb"))
{
    OleDbCommand command = new OleDbCommand(sql2, connection);
    connection.Open();
    command.ExecuteNonQuery();
    connection.Close();
}

//tulostetaan vahvisus poistosta
LblVahvistus.Text = "Tiedot on poistettu tietokannasta!";
```

#### 4.3.5 Poista laite/laitteen lisätiedot

Tilahakutulossivulla jokaisen laitteen kohdalla on painike lisätietojen tarkastelua varten. Lisätietopaneelissa (kuvio 14) näytetään laitteen kaikki tiedot tekstikentissä. Tekstikentistä ja selitekentistä on näkyvissä vain kulloisenkin laitteen tarvitsemat kentät. Tämä on toteutettu lisätietopainikkeiden taakse asettamalla tekstikenttien Visible-ominaisuuden (Esimerkki 8) arvoksi joko "true" tai "false". Esimerkissä on kyse tulostimen lisätietotekstikentistä, jossa turhat tekstikentät on asetettu nä-

kymättömiksi. Kaikkien laitteiden ja käyttäjien lisätiedot näytetään samassa paneelissa, joten useamman laitteen lisätietoja ei voida tarkastella samanaikaisesti. Laitteet poistetaan tietokannasta poistopainikkeella.



The image shows a web-based configuration interface for a printer, set against a light green background. It contains five labeled input fields and a button. The fields are: 'Ip' with the value '2.3.4.5', 'Malli' with 'Brother', 'Serial' with '436365', 'Kasetti' with 'eee', and 'Tila' with 'Huone2'. Below these fields is a button labeled 'Tallenna tulostimen muutokset'.

Ip	2.3.4.5
Malli	Brother
Serial	436365
Kasetti	eee
Tila	Huone2

Tallenna tulostimen muutokset

Kuvio 14. Tulostimen lisätietopaneeli.



Esimerkki 8. Tekstikenttien, otsikoiden ja painikkeiden asettaminen näkyväksi/näkymättömäksi.

```

Panel1.Visible = true;
TbOs.Visible = false;
TbLeasing.Visible = false;
LblOs.Visible = false;
LblLeasing.Visible = false;
TbLisaTietokone.Visible = false;
LblTietokone.Visible = false;
TbIp.Visible = true;
LblIp.Visible = true;
TbKasetti.Visible = true;
LblKasetti.Visible = true;
LblKovo.Visible = false;
TbLisaKovo.Visible = false;
LblRam.Visible = false;
TbLisaRam.Visible = false;
LblCpu.Visible = false;
TbLisaCpu.Visible = false;
TbLisaTila.Visible = true;
LblTila.Visible = true;

BtnPoistaTulostin.Visible = true;
BtnPoistaTietokone.Visible = false;
BtnPoistaNaytto.Visible = false;
BtnPoistaDvdvhs.Visible = false;
BtnPoistaVideotykki.Visible = false;

BtnTallennaTietokone.Visible = false;
BtnTallennaNaytto.Visible = false;
BtnTallennaTulostin.Visible = true;
BtnTallennaDvdvhs.Visible = false;
BtnTallennaVideotykki.Visible = false;

```

#### 4.3.6 Muokkaa laitteen/käyttäjän tietoja

Laite- ja käyttäjätietoja voi muokata tarkastelemalla ensin lisätietoja. Uudet tiedot muokataan suoraan lisätietopaneelin tekstikenttiin vanhojen tietojen päälle. Esimerkkinä on käytetty tulostimen tietojen muokkaamista. Muokatut tiedot tallennetaan tietokantaan update-lauseella (Esimerkki 9).

### Esimerkki 9. Tulostintietojen muokkaaminen update-lauseella.

```
protected void BtnTallennaTulostin_Click(object sender, EventArgs e)
{
    string sql = @"UPDATE Tulostin SET TulostinMalli='" + TbMalli.Text + "', "
        + "TulostinSijainti='" + TbLisaTila.Text + "', "
        + "TulostinIp='" + TbIp.Text + "', Kasetti='" + TbKasetti.Text + "', "
        + "TulostinSerial='" + TbSerial.Text + "' "
        + "WHERE TulostinID LIKE '" + TbTulostinID.Text + "'";

    using (OleDbConnection connection = new OleDbConnection
        ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=G:\ont\Ohjelma\LaiHa\Laitteet5.mdb"))
    {
        OleDbCommand command = new OleDbCommand(sql, connection);
        connection.Open();
        command.ExecuteNonQuery();
        connection.Close();
    }

    LblVahvistus2.Text = "Muutokset on tallennettu tietokantaan!";
}
```

#### 4.3.7 Lisää käyttäjä

Uuden käyttäjän tiedot lisätään Lisää uusi käyttäjä -painikkeella, jolloin ohjelmassa kontrolli siirtyy Lisää\_käyttäjä-lomakkeelle (kuvio 15) Response.Redirect-metodin avulla. Lomakkeella uudelle käyttäjälle syötetään nimi ja valitaan tila sekä tietokone alasvetovalikoista. Tila-alasvetovalikko täytetään sivun PageLoad-osiossa Tila- taulusta löytyvillä tilanimillä. PageLoad-osiossa haetaan tilanimen lisäksi tietokoneet tila-alasvetovalikossa ensimmäisenä näkyvälle tilalle. Näin tietokoneet sisältävä alasvetovalikko ei ole tyhjä tultaessa lisäyssivulle. Tietokoneet hakevassa sql-lauseessa (Esimerkki 10) tietokoneen sijainniksi on asetettu tila-alasvetovalikon sen hetkinen valinta.

Käyttäjän nimi

Työhuone

Tietokone

Kuvio 15. Uuden käyttäjän lisäyssivu.

Esimerkki 10. Sql-lause tietokoneiden hakemista varten.

```
//tietokoneen sijainniksi tulee tila-alasvetovalikon valinta
string sijainti = DropTila.SelectedItem.Text;
string sql1 = "Select TietokoneNimi from Tietokone "
    + "where TietokoneSijainti =' " + sijainti + "'";
```

Loppukäyttäjä valitsee työhuoneen tila-alasvetovalikosta. Sovellus hakee Selecte-  
dIndexChanged-tapahtumalla (Esimerkki 11) tietokone-alasvetovalikkoon tilassa  
olevat tietokoneet. Sql-lauseessa määritetään tietokoneen sijainniksi tila-  
alasvetovalikosta valitun tilan nimi.

### Esimerkki 11. SelectedIndexChanged-tapahtuma.

```
protected void DropTila_SelectedIndexChanged(object sender, EventArgs e)
{
    //tyhjätään alasvetovalikko heti alussa,
    //ettei edellisen valinnan tulokset jää näkyviin
    DropTietokone.Items.Clear();

    //TietokoneSijainti saa arvokseen tila-alasvetovalikosta
    //valitun tilanimen
    string sijainti = DropTila.SelectedItem.Text;
    string sql = "Select TietokoneNimi from Tietokone "
        + "where TietokoneSijainti = '" + sijainti + "'";

    using (OleDbConnection connection = new OleDbConnection
        ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=G:\\ont\\Ohjelma\\LaiHa\\Laitteet5.mdb"))
    {
        OleDbCommand command = new OleDbCommand(sql, connection);
        connection.Open();

        OleDbDataReader reader = command.ExecuteReader();

        while (reader.Read())
        {
            string tietokone = (string)reader["TietokoneNimi"];
            DropTietokone.Items.Add(tietokone);
        }
        reader.Close();
        connection.Close();
    }
}
```

Tila-alasvetovalikon AutoPostBack-ominaisuuden arvoksi on asetettu "true", koska palvelimelle on lähetettävä pyyntö aina kun loppukäyttäjä tekee eri valinnan alasvetovalikosta. Jokaisella pyyntökerralla tapahtuma käsitellään palvelimella eli suoritetaan koodi, jolla tietokone-alasvetovalikko täytetään. (Cuckoo 2004.)

Kaikki tiedot ovat pakollisia, mutta käytännössä ainoastaan käyttäjänimi voisi jäädä puuttumaan, sillä alasvetovalikoissa on aina valittuna jotkin arvot. Onnistuneen lisäyksen jälkeen näytetään vahvistusviesti.

#### 4.3.8 Lisää laite

Jokaiselle laitteelle on oma lisäyspainikkeensa, esim. Lisää uusi tietokone -painikkeen takaa päästään lisäämään ainoastaan tietokoneita. Lisäyspainikkeet ovat aina nähtävillä sivun alalaidassa. Tässä on käytetty esimerkkinä uuden tietokoneen lisäämistä.

Sivun vasempaan laitaan avautuu Lisää\_tietokone-lomake, jossa on tekstikentät tietojen syöttämistä varten (kuvio 16). Pakollisia tietoja uutta tietokonetta lisättäessä ovat tietokoneen nimi ja ip-osoite. Ohjelma tarkistaa pakolliset kentät RequiredFieldValidator-kontrollilla. Mikäli loppukäyttäjä jättää kyseiset kohdat tyhjiksi, näytetään hänelle ValidationSummary-kontrollin (kuvio 17) avulla yhteenveto kaikista puuttuvista tiedoista, eikä ohjelma tallenna uutta tietokonetta ennen kuin pakolliset tiedot on täytetty. Tietokannassa jokainen kenttä on määritetty pakolliseksi tiedoksi, mutta niille on asetettu oletusarvo. Näin loppukäyttäjän ei tarvitse heti syöttää vähemmän tärkeitä tietoja, kuten sarjanumeroa päästäkseen eteenpäin.

Tietokoneen nimi	<input type="text"/>	Näytön malli	<input type="text"/>
Ip	<input type="text"/>	Näytön serial	<input type="text"/>
Sijainti	<input type="text" value="Huone1"/>	Näytön leasing	<input type="text"/>
Cpu	<input type="text"/>		
Ram	<input type="text"/>		
Kovalevy	<input type="text"/>		
Käyttöjärjestelmä	<input type="text"/>		
Malli	<input type="text"/>		
Serial	<input type="text"/>		
Leasing	<input type="text"/>		
<input type="button" value="Lisää"/>			

Kuvio 16. Uuden tietokoneen lisäyssivu.

The image shows a validation summary form with a light green background. It contains the following fields and values:

- Tietokoneen nimi: [Empty text box] \*
- Ip: 192.168.1.20
- Sijainti: Huone1 (dropdown menu)
- Cpu: [Empty text box]
- Ram: [Empty text box]
- Kovalevy: Maxtor 400 Gb
- Käyttöjärjestelmä: Win 7
- Malli: Osborne
- Serial: [Empty text box]
- Leasing: [Empty text box]

At the bottom, there is a red asterisk and the text "Pakollinen tieto!" (Required information!).

Kuvio 17. ValidationSummary.

Lisäyssivulla on alasvetovalikko tilanimille, josta loppukäyttäjä voi poimia tilan, johon lisättävä tietokone sijoitetaan. Alasvetovalikon täyttäminen tapahtuu sivun PageLoad-osiossa while-toistorakennetta käyttäen. Tilanimiä varten on luotu nimi-muuttuja, johon tilanimet poimitaan yksi kerrallaan ja muuttujan saamat arvot lisätään alasvetovalikkoon. Ohjelma toistaa while-rakennetta (Esimerkki 12), kunnes kaikki tilanimet on haettu.

Esimerkki 12. While-rakenne.

```
//reader-lukija lukee tilanimet, kunnes kaikki on käyty läpi
while (reader.Read())
{
    //tilanimet lisätään nimi-muuttujaan
    string nimi = (string)reader["TilaNimi"];
    //alasvetovalikko täytetään tilanimillä,
    //jotka reader on käynyt lukemassa tietokannasta
    DropTila.Items.Add(nimi);
}
```

Varsinainen lisäyskoodi on Lisää-painikkeen takana. Lisäämiseen on käytetty sql-kielen insert-lausetta. Arvot poimitaan tekstikentistä, paitsi tilanimet poimitaan alasvetovalikosta. Suluissa kerrottujen lomakekomponenttien täytyy olla täsmäl-

leen samassa järjestyksessä, kuin insert-lauseessa luetellut taulun sarakkeet. Muutoin tiedot tallentuisivat väärin sarakkeisiin.

Lisättävälle tietokoneelle on samalla mahdollista määrittää yksi näyttö. Tätä varten Lisää\_tietokone-lomakkeessa on tekstikentät myös näytön tietojen syöttämiseksi. If-lauseella (Esimerkki 13) tarkistetaan, onko TbNayttoMalli-tekstikenttään syötetty tietoa, eli onko loppukäyttäjä syöttänyt näytölle mallin. Mikäli tekstikenttä ei ole tyhjä, järjestelmä tietää lisätä näytön tietokantaan. Tietokannassa on määritetty, että jokaisen näytön tulee olla yhteydessä tiettyyn tietokoneeseen. Tässä tapauksessa näytölle poimitaan tietokone lisättävän tietokoneen TbTietokone-tekstikentästä, johon syötetään tietokoneen nimi. Näytön tekstikenttiä ei ole validoitu. Yhdelle tietokoneelle voidaan määrittää useampi näyttö, joten uuden näytön lisäämiselle on erikseen lisäyspainike. Tietokoneelle ei ole pakko heti lisätä näyttöä, vaan sen voi tehdä myöhemmin.

**Esimerkki 13.** Tarkistetaan if-lauseella, onko tekstikenttään syötetty tietoa.

```
//jos tekstikenttä ei ole tyhjä, suoritetaan if:n sisällä olevat lauseet
if (TbNayttoMalli.Text != "")
{
    //tällä lauseella lisätään näyttö tietokoneelle, jota ollaan juuri lisäämässä
    string sql1 = "INSERT INTO Naytto (NayttoMalli, NayttoSerial, NayttoLeasing, TietokoneNimi) "
        + "VALUES ('" + TbNayttoMalli.Text + "', '" + TbNayttoSerial.Text + "', '" + TbNayttoLeasing.Text + " "
        + "'" + TbTietokone.Text + "')";

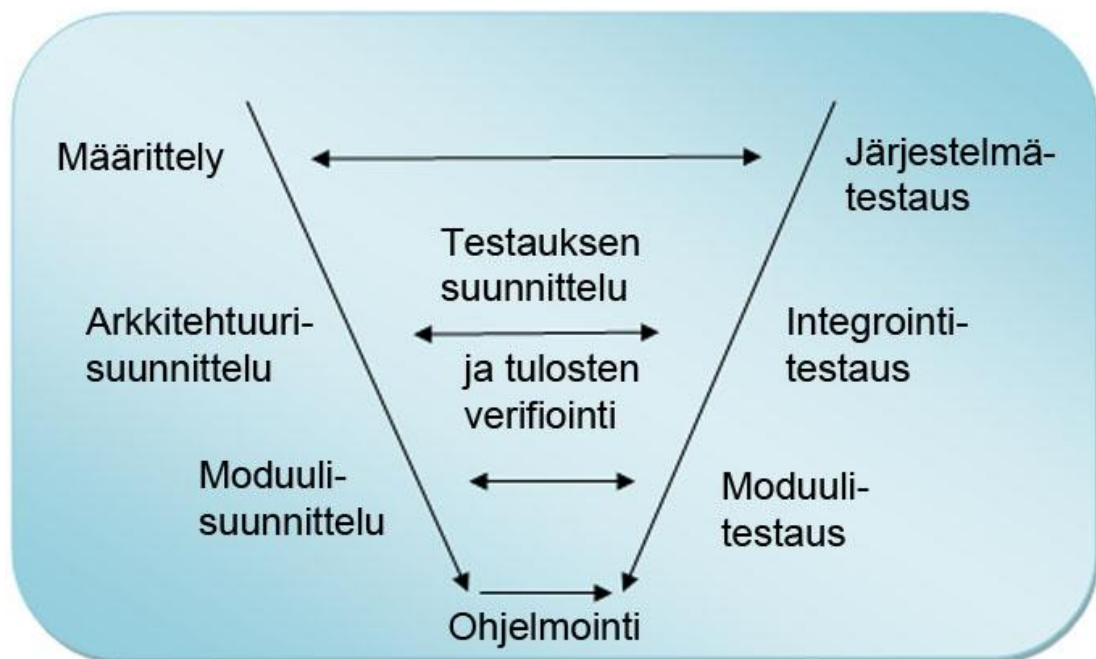
    using (OleDbConnection connection = new OleDbConnection
        ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=G:\ont\Ohjelma\LaiHa\Laitteet5.mdb"))
    {
        OleDbCommand command = new OleDbCommand(sql1, connection);
        connection.Open();
        command.ExecuteNonQuery();
        connection.Close();
    }
    LblVahvistus.Text = "Tiedot on lisätty tietokantaan!";
}
```

Kun pakolliset tiedot on täytetty, uusi tietokone ja mahdollisesti sille lisätty näyttö tallennetaan tietokantaan. Loppukäyttäjälle tulostetaan tästä vahvistusviesti LblVahvistus-muuttujaan. Jos loppukäyttäjä päättää olla lisäämättä uutta laitetta, lisäyssivulta poistutaan logoa painamalla etusivulle.



## 5 SOVELLUKSEN TESTAUS

Ohjelmistoa testattaessa siitä etsitään suunnitelmallisesti virheitä. Testaus tarkoittaa lähes mitä tahansa kokeilemista. Ohjelmistoa testattaessa edetään ns. V-mallin mukaisesti (kuvio 18), joka sisältää moduuli-, integrointi- ja järjestelmätestauksen. Järjestelmätestaus suunnitellaan määrittelyvaiheessa, moduulitestaus moduulisuunnitteluvaiheessa ja integrointitestaus suunnitteluvaiheessa. Tuloksia verrataan niitä vastaaviin dokumentteihin oikeellisuuden toteamiseksi. (Haikala & Märijärvi 2004, 284, 288, 289.)



Kuvio 18. V-malli.

Lähde: Haikala & Märijärvi 2004, 289.

Moduulitestauksessa testataan yksittäistä moduulia, joka koostuu yleensä 100–1000 koodirivistä. Moduulitestausta suoritettiin koko ohjelmoinnin ajan tekijän toimesta. Heti moduulin valmistuttua sitä testattiin ja virheet korjattiin. Samoin integrointitestausta suoritettiin koko sovelluksen rakentamisen ajan. Integrointitestauksessa yhdisteltiin moduuleja ja testattiin niiden keskinäistä toimivuutta. Erityisesti kiinnitettiin huomiota käyttöliittymän käytettävyyteen.

Järjestelmätestauksen testitapaukset on valittu ns. lasilaatikkotestauksen perusteella, jossa käytetään hyväksi tietoa järjestelmän toteutuksesta. Järjestelmätestauksista suoritettiin tulevien loppukäyttäjien toimesta. Myös sovelluksen tekijä suoritti



järjestelmätestausta sovelluksen rakentamisen loppuvaiheessa. Kaikki testitapaukset dokumentoitiin, joista on esitetty yksi esimerkkinä (taulukko 2). Loput testitapaukset löytyvät tämän työn lopusta liitteestä 3. Testauksessa todettiin sovelluksen toimivan toiminnallisen määrittelyn kuvaamalla tavalla, eikä sovelluksen toiminnassa ilmennyt ongelmia.

Taulukko 2. Testitapaus: käyttäjätietojen muokkaus.

Testattava kohde	Käyttäjätietojen muokkaus
Testauksen kuvaus	Testataan käyttäjätietojen muokkaamista järjestelmässä.
Testauksen tulos	Testaus onnistui.
Testaaja	Kimmo Puumala
Testaajan kommentti	Lisätiedoissa olisi hyvä näyttää myös tietokoneen nimi, jota muokkaamalla käyttäjälle voisi vaihtaa toisen koneen.
Tekijän kommentti	Käyttäjän lisätietoihin lisättiin alasvetovalikko, josta käyttäjälle saa vaihdettua toisen koneen.

## 6 ARVIOINTI JA JATKOKEHITYS

Opinnäytetyön tarkoituksena oli suunnitella ja rakentaa Seinäjoen ammattikorkeakoulun Sosiaali- ja terveys alan yksikölle web-sovellus atk-laitteiden hallintaan. Sovelluksen päätavoitteena oli mahdollistaa tietojen hakeminen eri perustein sekä tietojen lisääminen, poistaminen ja muokkaaminen. Sovelluksen tarkoituksena oli nopeuttaa ja selkeyttää päivittäistä työskentelyä laitetietojen parissa.

Sovelluksen sovelluskehittimeksi valittiin Visual Studio 2008, joka oli tuttu aiemmilta kursseilta. Myös ohjelmointikieleksi valittu C# oli ennestään tuttu käydyltä kursseilta. Täten tekninen toteutus sujui lähes ongelmitta. Perustietämys relaatiotietokannoista ja sql-kyselykielestä olivat avuksi tietokantaa luotaessa sekä yhteyksien ja hakujen rakentamisessa sovellukseen. Hankalissa tilanteissa apua löytyi runsaasti internetistä. Sieltä löytyi hyvin koodiesimerkkejä, joista otettiin mallia tiettyjä toimintoja rakennettaessa.

Sivuston ulkoasu on selkeä ja yhtenäinen. Toiminnot on helposti löydettävissä sivustolta ja toimeksiantajan mielestä ne on helppo suorittaa. Sivusto informoi loppukäyttäjää ongelmatilanteissa ja opastaa eteenpäin. Käyttäjäystävällisyys saavutettiin minimoimalla selaimelle tulostuvat virheilmoitukset.

Sovellukseen kaavailtiin raportointimahdollisuutta, jossa loppukäyttäjä voisi tulostaa raportin tietyistä tiedoista. Tietokantaan mietittiin tallennettavaksi myös oppilaitoksen henkilökunnan puhelimet. Aikataulun puitteissa näitä ei voitu kuitenkaan toteuttaa. Käyttäjänimen perusteella suoritettua haun hakutuloksissa mietittiin näytettäväksi käyttäjän kuva käyttäjänimen vieressä. Tämä olisi ollut hyvä varsinkin atk-tuessa työskentelevälle harjoittelijalle. Kuvien käyttöön sovelluksessa olisi tarvittu yksikön johtajan lupa, mutta lupaa ei kysytty ja asiaan ei palattu enää sovelluksen rakentamisen aikana.

Aikataulu riitti hyvin sekä sovelluksen toteutukseen että dokumentointiosuuden kirjoittamiseen. Työssä ei ole dokumentoitu käyttöönottoa. Käyttöönoton suorittaa tekijä yhdessä atk-tukihenkilöiden kanssa. Tekijä opastaa sovelluksen käytössä ja antaa käyttöohjeen.

Opinnäytetyön tekeminen antoi hyvän kuvan ohjelmistokehittäjän työstä ja ohjelmistokehitysprosessista. Työtä tehdessä huomasin, miten tärkeitä asiakasvaatimusten kartoitus ja toimeksiantajan kanssa käydyt sovelluksen kehittämiskeskustelut ovat. Havaitsin, että sovelluksen kunnollinen suunnittelu auttoi saavuttamaan toivotunmukaisen sovelluksen. Tämä mahdollisti, että testauksen jälkeen ei havaittu tarvetta isoille korjauksille tai muutoksille. Vaikka työn toteuttamiseen olikin hyvät valmiudet, tuli silti eteen muutamia ongelmatilanteita, joihin löydetyistä ratkaisuista oppi uutta.

## LÄHTEET

- Cuckoo, J. J. 2004. AutoPostBack - What it is and how it works? [WWW-artikkeli]. SpiderWorks Technologies Pvt Ltd. [Viitattu 18.1.2011]. Saatavissa: <http://www.dotnetspider.com/resources/189-AutoPostBack-What-How-works.aspx>
- Eriksson, H-E. & Penker, M. 2000. UML. Kääntänyt Tarmo Toikkanen. Jyväskylä: Edita.
- Haikala, I. & Märijärvi, J. 2004. Ohjelmistotuotanto. 10. painos. Helsinki: Talentum Media Oy.
- Hernandez, M. J. 2000. Tietokannat: suunnittelu käytännössä. Jyväskylä: Gummerus.
- Hovi, A. 2004. Sql-opas. Jyväskylä: Docendo.
- Hovi, A., Huotari, J. & Lahdenmäki, T. 2005. Tietokantojen suunnittelu ja indeksointi. Porvoo: WS Bookwell.
- Javascript confirm. [WWW-dokumentti]. Tizag. Erack Network. [Viitattu 22.1.2011]. Saatavissa: <http://www.tizag.com/javascriptT/javascriptconfirm.php>
- Järvinen, P. & Järvinen, A. 2000. Tutkimusmetodeista. Tampere: Opinpajan kirja.
- Lahtonen, T. 2004. SQL. 5. painos. Vantaa: Docendo.
- Larman, C. 2004. Applying UML and patterns: an introducing to object-oriented analysis and design and iterative development. 3. painos. Upper Saddle River: Pearson Education, Inc.
- ListItem Constructor. [WWW-dokumentti]. Microsoft Corporation. [Viitattu 20.1.2011]. Saatavissa: <http://msdn.microsoft.com/en-us/library/cf4h1yxc.aspx>
- Pohjonen, R. 2002. Tietojärjestelmien kehittäminen. 2. painos. Jyväskylä: Docendo.
- Webopedia. [WWW-dokumentti]. [Viitattu 25.1.2011]. Saatavissa: <http://www.webopedia.com/>

## LIITTEET

Liite 1: Käyttötapauskuvaukset

Liite 2: Sovelluksen sekvenssikaaviot

Liite 3: Sovelluksen testausdokumentit

Liite 4: Käyttöohje

**LIITE 1. Käyttötapauskuvaukset**

Käyttötapaus: Laitteen tai käyttäjän lisäys	
Aktori	Loppukäyttäjä
Frekvenssi	Kun halutaan lisätä uusi laite tai käyttäjä järjestelmään
Esiehdot	Järjestelmä on toimintakykyinen
Kuvaus	Uusi laite tai käyttäjä lisätään tietokantaan
Lopputulos	Uusi laite tai käyttäjä on tietokannassa
Poikkeukset	-

Käyttötapaus: Laitteen tai käyttäjän poisto	
Aktori	Loppukäyttäjä
Frekvenssi	Kun halutaan poistaa laite tai käyttäjä järjestelmästä
Esiehdot	Laite tai käyttäjä on tietokannassa
Kuvaus	Laite tai käyttäjä poistetaan tietokannasta
Lopputulos	Laite tai käyttäjä on poistettu tietokannasta
Poikkeukset	Laitetta tai käyttäjää ei ole tietokannassa

Käyttötapaus: Tietojen muokkaus	
Aktori	Loppukäyttäjä
Frekvenssi	Kun halutaan muokata laitteiden tai käyttäjien tietoja
Esiehdot	Laite tai käyttäjä on tietokannassa
Kuvaus	Laitteen tai käyttäjän tiedot muokataan tietokantaan
Lopputulos	Laitteella tai käyttäjällä on päivitetty tiedot
Poikkeukset	Laitetta tai käyttäjää ei ole tietokannassa

Käyttötapaus: Hakeminen alasvetovalikosta	
---	--

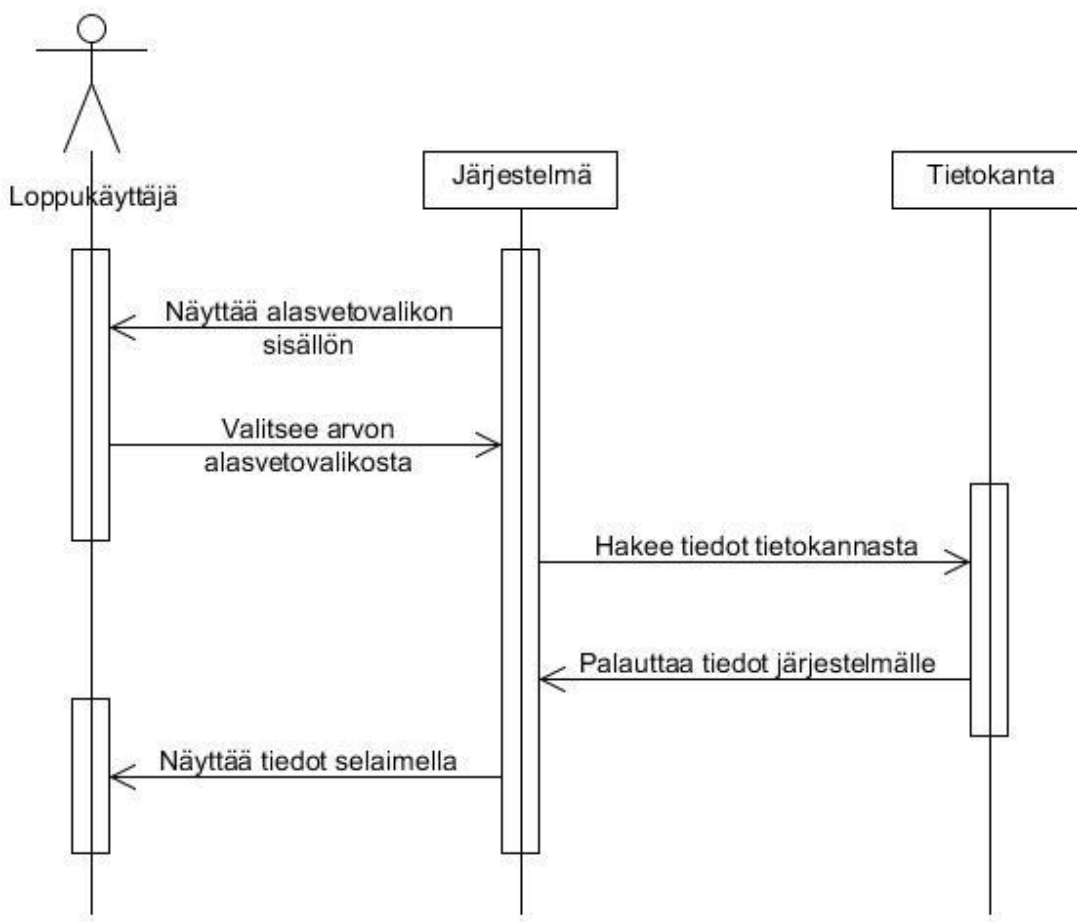
Aktori	Loppukäyttäjä
Frekvenssi	Kun halutaan hakea laitetietoja alasvetovalikoiden tarjonnasta valitsemalla.
Esiehdot	Järjestelmä on toimintakuntoinen
Kuvaus	Laitetiedot haetaan alasvetovalikosta valitun arvon perusteella
Lopputulos	Järjestelmä tulostaa laitetiedot loppukäyttäjälle
Poikkeukset	-

Käyttötapaus: Hakeminen hakusanalla	
Aktori	Loppukäyttäjä
Frekvenssi	Kun halutaan hakea laitetietoja kirjoittamalla hakusana
Esiehdot	Järjestelmä on toimintakuntoinen
Kuvaus	Laitetiedot haetaan kirjoittamalla hakusana ja valitsemalla oikea kategoria alasvetovalikosta (käyttäjänimi, tilanimi, ip-osoite).
Lopputulos	Järjestelmä tulostaa laitetiedot loppukäyttäjälle
Poikkeukset	Mikäli hakusana ja valittu kategoria eivät täsmää, järjestelmä tulostaa tyhjät tuloskentät käyttäjälle.

## LIITE 2. Sovelluksen sekvenssikaaviot

### Tietojen hakeminen alasvetovalikosta valitsemalla

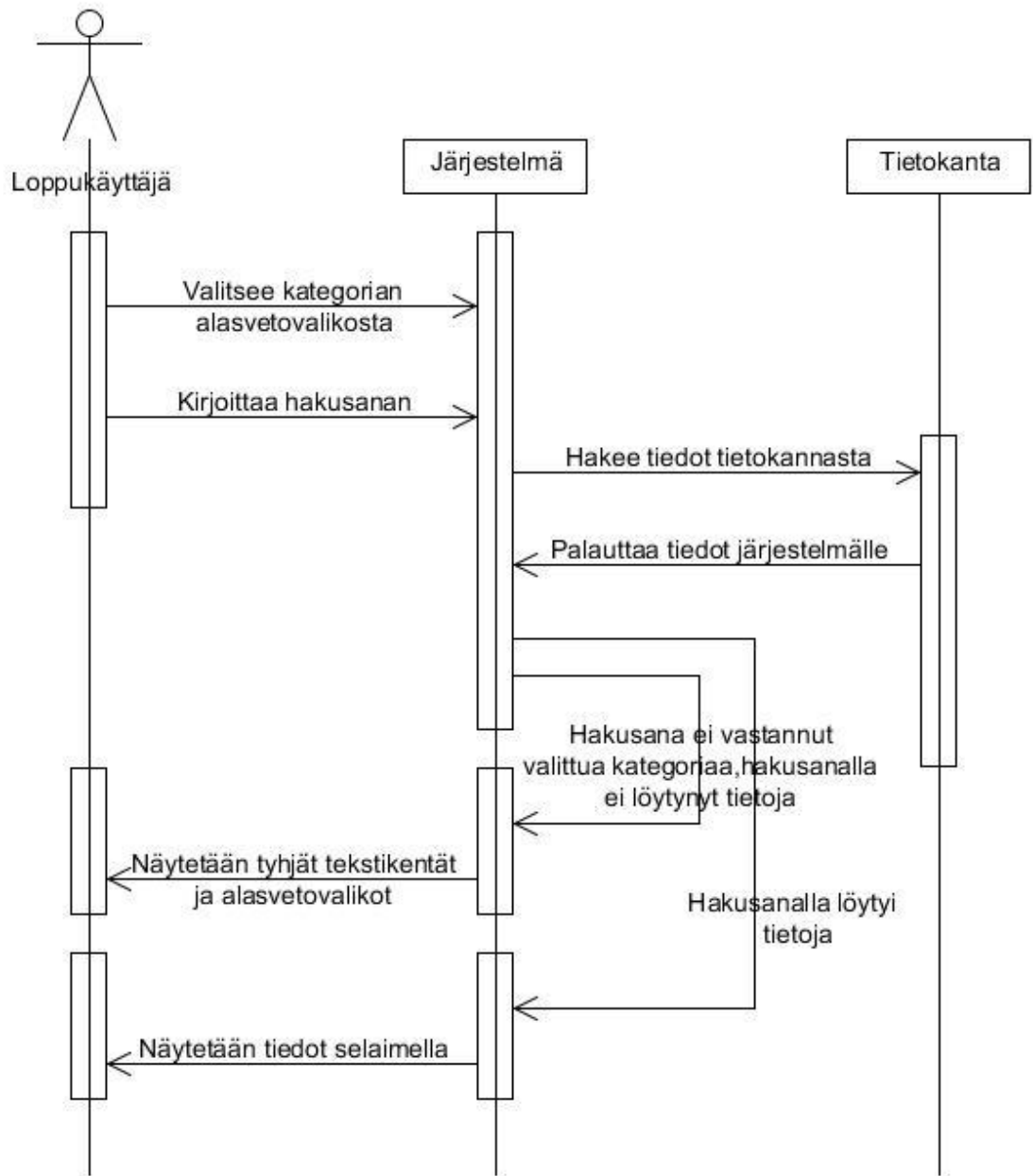
Järjestelmä hakee tiedot kuhunkin alasvetovalikkoon. Käyttäjä valitsee alasvetovalikosta arvon, jonka perusteella laitetiedot halutaan hakea. Järjestelmä hakee valittuun arvoon liittyvät laitetiedot tietokannasta ja tulostaa hakutulokset käyttäjän selaimelle.



### Tietojen hakeminen hakusanalla

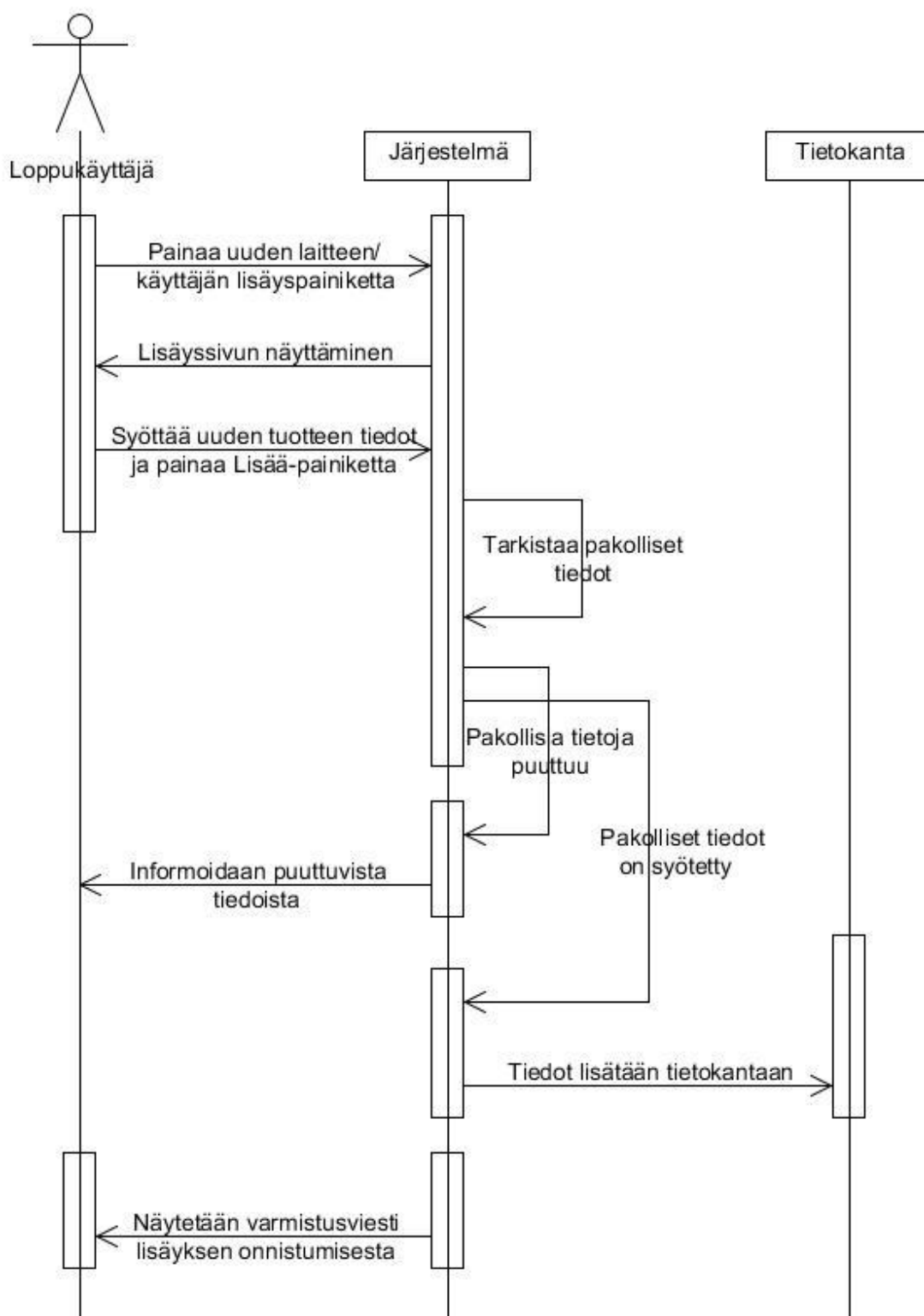
Käyttäjä valitsee kategorian alasvetovalikosta sen mukaan, suoritetaanko haku käyttäjänimen, ip-osoitteen vai tilanimen perusteella. Sen jälkeen käyttäjä kirjoittaa hakusanan ja painaa Hae tiedot- painiketta. Järjestelmä hakee tiedot tietokannasta. Jos hakusanan mukaista tietoa ei ole tietokannassa, järjestelmä tulostaa tyhjän hakutuloksen. Jos hakusana löytyy tietokannasta, siihen liittyvät laitetiedot tulostetaan käyttäjälle.





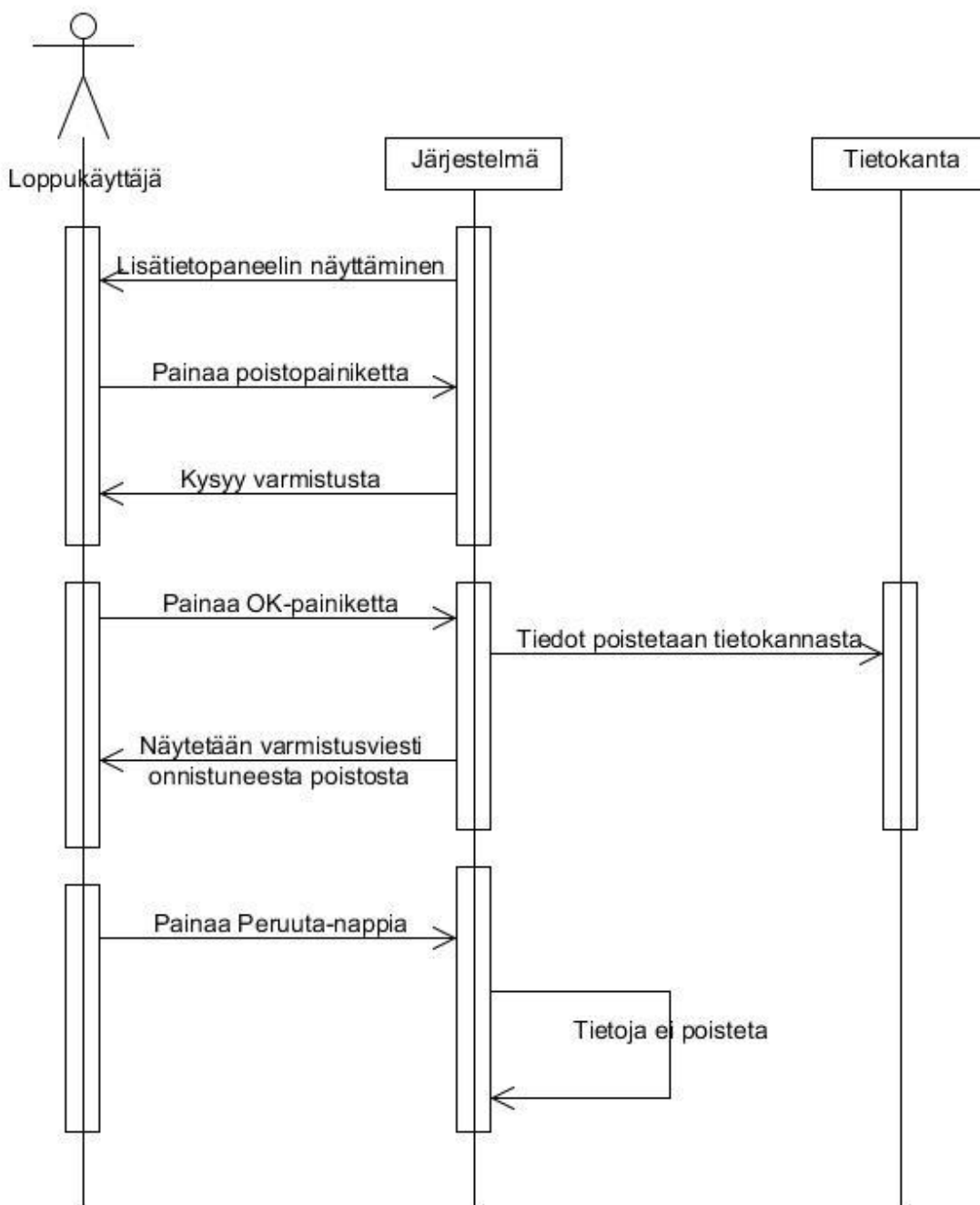
### Tietojen lisääminen

Käyttäjä painaa uuden laitteen/käyttäjän lisäyspainiketta, jolloin järjestelmä näyttää uuden laitteen/käyttäjän lisäyssivun. Käyttäjä syöttää uudet laite/käyttäjätiedot ja painaa Lisää-painiketta. Järjestelmä tarkistaa puuttuuko pakollisia tietoja. Jos tiedot ovat puutteelliset, järjestelmä tiedottaa siitä käyttäjälle. Jos käyttäjä on syöttänyt kaikki pakolliset tiedot, uusi laite/käyttäjä tallennetaan tietokantaan ja käyttäjälle tulostetaan varmistusviesti lisäyksen onnistumisesta.



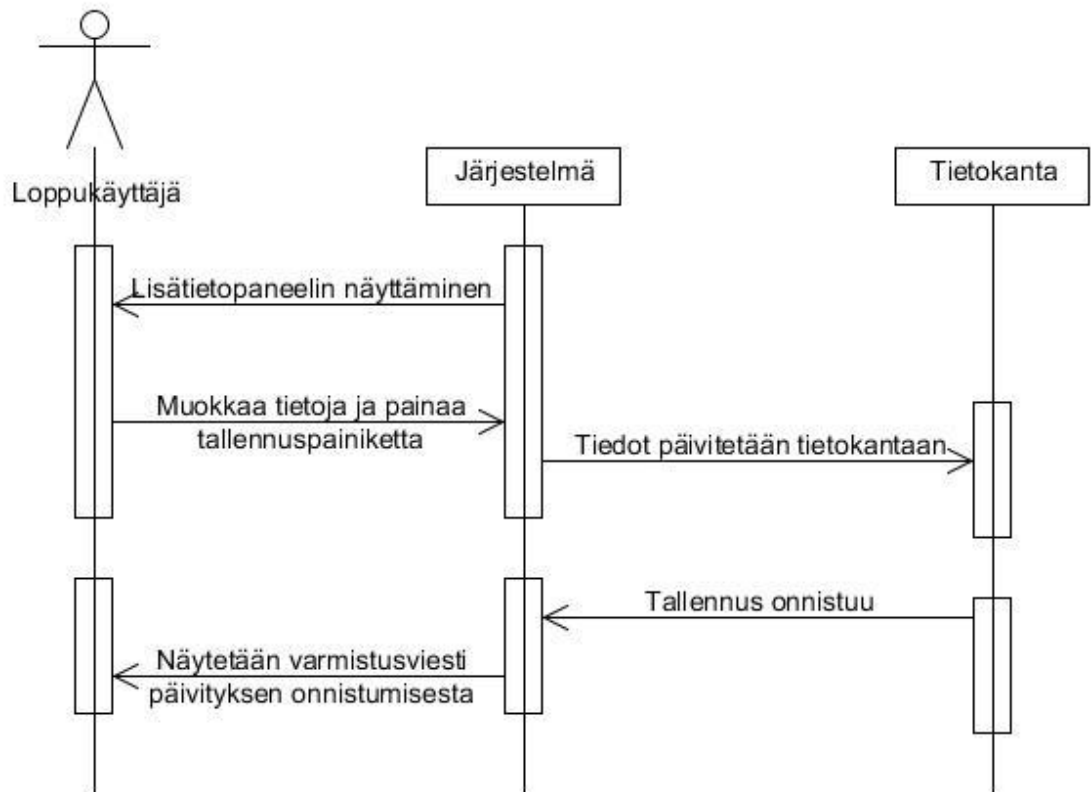
## Tietojen poistaminen

Järjestelmä näyttää laitteen/käyttäjän lisätietopaneelin. Käyttäjä painaa lisätietopaneelissa käyttäjän/tietokoneen nimen tai laitteen mallin kohdalla olevaa poistopainiketta. Järjestelmä näyttää varmistusikkunan, jossa kysytään varmistusta tietojen poistamisesta. Jos käyttäjä painaa OK-painiketta, tiedot poistetaan tietokannasta ja onnistuneesta poistosta näytetään varmistusviesti käyttäjälle. Mikäli käyttäjä painaa varmistusikkunan Peruuta-painiketta, tietoja ei poisteta tietokannasta.



## Tietojen muokkaaminen

Järjestelmä näyttää lisätietopaneelin. Käyttäjä muokkaa laite/käyttäjätietoja suoraan vanhojen tietojen päälle ja painaa tallennuspainiketta. Järjestelmä päivittää tiedot tietokantaan ja näyttää käyttäjälle onnistuneesta tallennuksesta varmistusviestin.



**LIITE 3. Sovelluksen testausdokumentit**

Testattava kohde	Uuden laitteen lisääminen.
Testauksen kuvaus	Testataan uuden laitteen lisäämistä järjestelmään.
Testauksen tulos	Testaus onnistui.
Testaaja	Kimmo Puumala
Testaajan kommentti	Näytönlisäyssivulla näytölle lisättävä tietokone tuottaa ongelmia, koska tietokoneen nimi pitäisi muistaa ulkoa, kun se kirjoitetaan tekstikenttään.
Tekijän kommentti	Tekstikentän sijaan luotiin alasvetovalikko, josta näytölle valitaan oikea tietokone, eikä nimeä näin ollen tarvitse muistella ulkoa.

Testattava kohde	Laitteen poistaminen.
Testauksen kuvaus	Testataan laitteen poistamista järjestelmästä.
Testauksen tulos	Testaus onnistui.
Testaaja	Kimmo Puumala

Testattava kohde	Laitetietojen muokkaaminen.
Testauksen kuvaus	Testataan laitetietojen muokkaamista järjestelmässä.
Testauksen tulos	Testaus onnistui.
Testaaja	Kimmo Puumala
Testaajan kommentti	Tilat voisi laittaa alasvetovalikkoon, josta laitteen sijaintia voisi vaihtaa.
Tekijän kommentti	Laitteiden lisätietosivuille lisättiin alasvetovalikko tekstikentän tilalle, jossa näytetään kaikki tilanimet.

Testattava kohde	Uuden käyttäjän lisääminen.
Testauksen kuvaus	Testataan uuden käyttäjän lisäämistä järjestelmään.
Testauksen tulos	Lisäys onnistui.
Testaaja	Petri Keskinen

Testattava kohde	Käyttäjän poistaminen
Testauksen kuvaus	Testataan käyttäjän poistamista järjestelmästä.
Testauksen tulos	Testaus onnistui.
Testaaja	Kimmo Puumala

Testattava kohde	Käyttäjätietojen muokkaus
Testauksen kuvaus	Testataan käyttäjätietojen muokkaamista järjestelmässä.
Testauksen tulos	Testaus onnistui.
Testaaja	Kimmo Puumala
Testaajan kommentti	Lisätiedoissa olisi hyvä näyttää myös tietokoneen nimi, jota muokkaamalla käyttäjälle voisi vaihtaa toisen koneen.
Tekijän kommentti	Käyttäjän lisätietoihin lisättiin alasvetovalikko, josta käyttäjälle saa vaihdettua toisen koneen.

Testattava kohde	Haku alasvetovalikoista
Testauksen kuvaus	Testataan laitetietojen hakemista alasvetovalikoiden kautta.
Testauksen tulos	Testaus onnistui.
Testaaja	Kimmo Puumala

Testattava kohde	Haku hakusanalla
Testauksen kuvaus	Testataan laitetietojen hakemista hakusanalla.
Testauksen tulos	Testaus onnistui.
Testaaja	Kimmo Puumala

Testattava kohde	Lisätietojen tarkastelu
Testauksen kuvaus	Testataan lisätietojen näyttämistä laite- ja käyttäjätietojen yhteydessä.
Testauksen tulos	Testaus onnistui.
Testaaja	Kimmo Puumala

## LIITE 4. Käyttöohje

Laitehallinta  
**Laita**



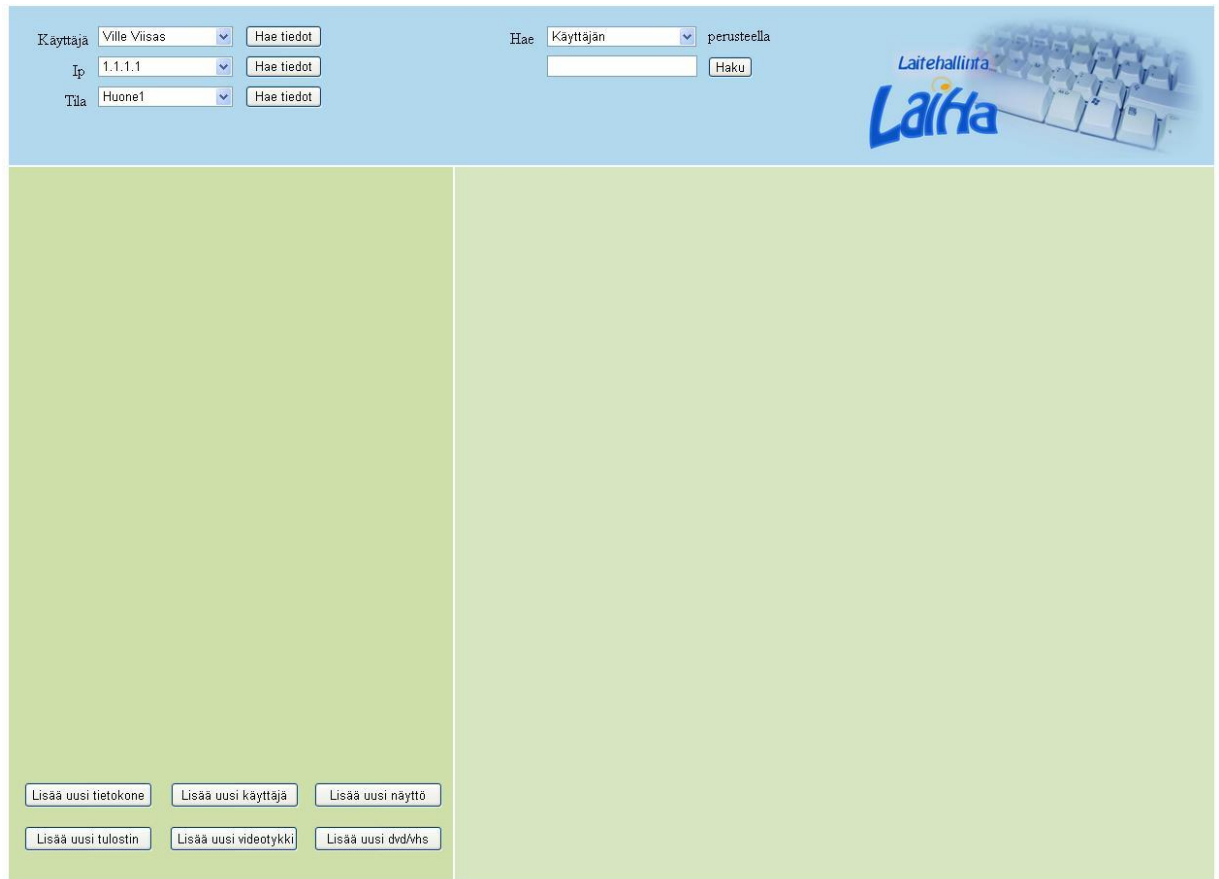
**SISÄLLYS**

1. Sivuston avaaminen.....	3
2. Etusivulle siirtyminen .....	3
3. Haku alasvetovalikoiden kautta.....	4
4. Haku hakusanalla .....	5
5. Lisätietojen tarkastelu .....	6
6. Tietojen lisääminen .....	8
7. Tietojen poistaminen .....	11
8. Tietojen muokkaaminen .....	15



## 1. SIVUSTON AVAAMINEN

Sivustolle pääsee kirjoittamalla palvelimen selaimen osoiteriville <http://gaappi:88>. Osoitteesta avautuu sivuston etusivu.



Käyttäjä Ville Viisas Hae tiedot

Ip 1.1.1.1 Hae tiedot

Tila Huone1 Hae tiedot

Hae Käyttäjän perusteella

Haku

Laittehallinta  
**LaiHa**

Lisää uusi tietokone Lisää uusi käyttäjä Lisää uusi näyttö

Lisää uusi tulostin Lisää uusi videotykki Lisää uusi dvd/hrs

## 2. ETUSIVULLE SIIRTYMINEN

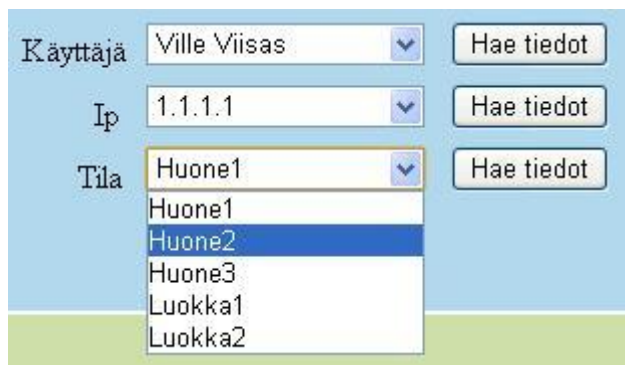
Etusivulle siirrytään klikkaamalla sivuston oikeassa yläaidassa olevaa logoa.



### 3. HAKU ALASVETOVALIKOIDEN KAUTTA

Alasvetovalikot sijaitsevat sivuston vasemmassa ylälaudassa. Laitetietoja haetaan joko käyttäjänimen, ip-osoitteen tai tilanimen perusteella, joille on omat alasvetovalikkonsa.

Haku tapahtuu valitsemalla alasvetovalikosta haluttu arvo



Käyttäjä	Ville Viisas	Hae tiedot
Ip	1.1.1.1	Hae tiedot
Tila	Huone1 Huone1 Huone2 Huone3 Luokka1 Luokka2	Hae tiedot

ja painamalla alasvetovalikon vieressä olevaa Hae tiedot- painiketta.



Tila	Huone2	Hae tiedot
------	--------	------------

Tämän jälkeen hakutulokset tulostuvat hakuosuuden alapuolelle



The screenshot shows a search interface on a light green background. It includes several input fields and buttons:

- Tila:** A text input field containing "Huone2".
- Käyttäjä:** A dropdown menu showing "Ville Viisas" with a blue arrow icon.
- Tietokone:** A dropdown menu showing "Villenkone" with a blue arrow icon, and a button labeled "Tietokoneen lisätiedot".
- Tulostin:** A dropdown menu showing "Brother" with a blue arrow icon, and a button labeled "Tulostimen lisätiedot".
- Näyttö:** A dropdown menu showing "Benq" with a blue arrow icon, and a button labeled "Näytön lisätiedot".
- Videotykki:** An empty text input field, and a button labeled "Videotykin lisätiedot".
- Dvd/Vhs:** An empty text input field, and a button labeled "Dvd/Vhs:n lisätiedot".
- Äänentoisto:** A text input field containing a hyphen "-", and a button labeled "Äänentoiston lisätiedot".

#### 4. HAKU HAKUSANALLA

Tietoja voi hakea myös kirjoittamalla täsmällinen hakusana. Tämä hakuosuus sijaitsee sivun ylälaudassa alasvetovalikkohaun oikealla puolella. Tässäkin tietoja voi hakea joko käyttäjänimen, ip-osoitteen tai tilanimen perusteella.

Haku tapahtuu valitsemalla alasvetovalikosta haettavan sanan kategoria



The screenshot shows a search interface on a light blue background. It includes a dropdown menu and a button:

- Hae:** A dropdown menu with "Käyttäjän" selected and highlighted in blue. The dropdown list shows "Käyttäjän", "Ip-osoitteen", and "Tilan".
- perusteella:** Text next to the dropdown menu.
- Hae tiedot:** A button to the right of the dropdown menu.

ja kirjoittamalla tekstikenttään hakusana sekä painamalla lopuksi Hae tiedot- painiketta.



Laitetiedot tulostuvat alapuolelle sivun vasempaan laitaan kuten alasvetovalikostakin haettaessa.

### **Huom!**

On varmistuttava, että kirjoitettu sana on mahdollista löytää valitusta kategoriasta. Muutoin hakutuloksena näytetään tyhjät tekstikentät. Esim. jos kategoriaksi on valittu käyttäjä, mutta hakusanaksi on kirjoitettu ip-osoite, ei haku tuota tällöin tulosta.

## **5. LISÄTIETOJEN TARKASTELU**

Hakutuloksissa näytetään vain tärkeimmät tiedot, mutta kaikkia tietoja on mahdollista tarkastella. Lisätietoja on mahdollista tarkastella ainoastaan käyttäjä- ja tilanimen perusteella tehtyjen hakujen yhteydessä. Jokaisen laitteen/käyttäjän kohdalla on lisätietopainike, jota painamalla näkyviin tulee laitteen/ käyttäjän kaikki tiedot.

Painetaan esim. Tulostimen lisätiedot- painiketta tulostimen mallin kohdalta.

Käyttäjä	<input type="text" value="Tauno Terävä"/>	<input type="button" value="Käyttäjän lisätiedot"/>
Työhuone	<input type="text" value="Huone1"/>	
Tietokone	<input type="text" value="Taunonkone"/>	<input type="button" value="Tietokoneen lisätiedot"/>
Cpu	<input type="text" value="fs"/>	
Ram	<input type="text"/>	
Kovalevy	<input type="text" value="sfsf"/>	
Tulostin	<input type="text" value="Hp100"/>	<input type="button" value="Tulostimen lisätiedot"/>
Tulostimen ip	<input type="text" value="1.2.3.4"/>	
Näyttö	<input type="text" value="34"/> ▼	<input type="button" value="Näytön lisätiedot"/>

Oikealle tulee näkyviin kaikki laitteen tiedot.

Käyttäjä	<input type="text" value="Tauno Terävä"/>	<input type="button" value="Käyttäjän lisätiedot"/>	
Työhuone	<input type="text" value="Huone1"/>		
Tietokone	<input type="text" value="Taunonkone"/>	<input type="button" value="Tietokoneen lisätiedot"/>	
Cpu	<input type="text" value="fs"/>		
Ram	<input type="text"/>		
Kovalevy	<input type="text" value="sfsf"/>		
Tulostin	<input type="text" value="Hp100"/>	<input type="button" value="Tulostimen lisätiedot"/>	
Tulostimen ip	<input type="text" value="1.2.3.4"/>		<input type="text" value="1.2.3.4"/>
Näyttö	<input type="text" value="34"/> ▼	<input type="button" value="Näytön lisätiedot"/>	
			<input type="text" value="Malli"/> <input type="text" value="Hp100"/>
			<input type="text" value="Serial"/> <input type="text" value="64t45"/>
			<input type="text" value="Kasetti"/> <input type="text" value="HpAkasetti"/>
			<input type="text" value="Tila"/> <input type="text" value="Huone1"/>
			<input type="button" value="Tallenna tulostimen muutokset"/>

## 6. TIETOJEN LISÄÄMINEN

Uuden laitteen/käyttäjän lisäyspainikkeet sijaitsevat sivun vasemmassa alalaidassa. Jokaista laitetta sekä käyttäjää varten on omat painikkeensa.

Painetaan esim. Lisää uusi tietokone- painiketta.



Avautuu lomake, johon uuden tietokoneen tiedot syötetään.

Tietokoneen nimi	<input type="text"/>	Näytön malli	<input type="text"/>
Ip	<input type="text"/>	Näytön serial	<input type="text"/>
Sijainti	<input type="text" value="Huone1"/>	Näytön leasing	<input type="text"/>
Cpu	<input type="text"/>		
Ram	<input type="text"/>		
Kovalevy	<input type="text"/>		
Käyttöjärjestelmä	<input type="text"/>		
Malli	<input type="text"/>		
Serial	<input type="text"/>		
Leasing	<input type="text"/>		
<input type="button" value="Lisää"/>			

Esimerkkitapauksessa tietokoneella voi samalla lisätä yhden näytön, mutta sen voi tehdä myöhemminkin painamalla Lisää uusi näyttö- painiketta ja valitsemalla tietokoneen alaspöytävalikosta.

Tiedot syötetään tekstikenttiin ja alaspöytävalikosta valitaan tila, johon tietokone halutaan sijoittaa. Kun tiedot on syötetty, painetaan Lisää-painiketta. Mikäli pakollisia tietoja puuttuu, kyseisten tekstikenttien viereen ilmestyy punainen tähti ja Lisää-painikkeen yläpuolelle tähden selite.

Tietokoneen nimi	<input type="text"/>	*
Ip	<input type="text" value="192.168.1.20"/>	
Sijainti	<input type="text" value="Luokka1"/>	▼
Cpu	<input type="text"/>	
Ram	<input type="text"/>	
Kovalevy	<input type="text" value="Maxtor 400 Gb"/>	
Käyttöjärjestelmä	<input type="text" value="Win 7"/>	
Malli	<input type="text" value="Osborne"/>	
Serial	<input type="text"/>	
Leasing	<input type="text"/>	
<div>• Pakollinen tieto!</div>		
<div>Lisää</div>		

Syötetään pakolliset tiedot ja jatketaan painamalla Lisää-painiketta. Kun lisäys onnistuu, Lisää-painikkeen viereen tulostetaan teksti ” Tiedot on lisätty tietokantaan!”, mikä tarkoittaa, että lisäys on onnistunut.



Tietokoneen nimi	<input type="text" value="OpenKone"/>
Ip	<input type="text" value="192.168.1.20"/>
Sijainti	<input type="text" value="Luokka1"/>
Cpu	<input type="text"/>
Ram	<input type="text"/>
Kovalevy	<input type="text" value="Maxtor 400 Gb"/>
Käyttöjärjestelmä	<input type="text" value="Win 7"/>
Malli	<input type="text" value="Osborne"/>
Serial	<input type="text"/>
Leasing	<input type="text"/>

Tiedot on lisätty tietokantaan!

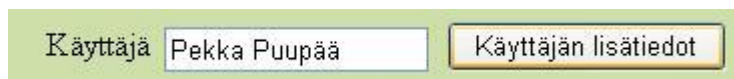
### Huom!

Mikäli tietojenlisäyslomake on jo esillä, mutta uutta laitetta/käyttäjää ei halutakaan lisätä, poistutaan siltä etusivulle painamalla logoa sivun oikeasta ylä laidasta. Tämä on ainoa tapa siirtyä pois tietojenlisäyslomakkeelta ja jatkaa sivuston käyttöä!

## 7. TIETOJEN POISTAMINEN

Tietoja voi poistaa ainoastaan yksitellen, ja tämä tapahtuu tarkastelemalla ensin laitteen/käyttäjän lisätietoja. Käyttäjiä voi poistaa ainoastaan käyttäjänimen perusteella tehtyjen hakujen yhteydessä, ja laitteita voi poistaa tilanimen perusteella tehtyjen hakujen yhteydessä.

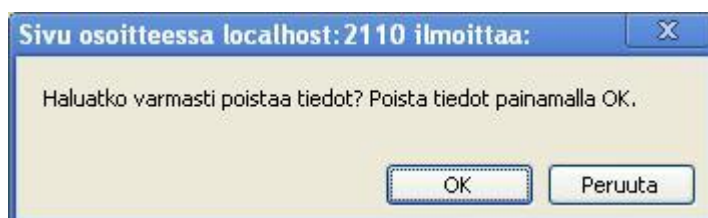
Käyttäjiä poistetaan suorittamalla ensin käyttäjänimen perusteella tehty haku ja painamalla hakutulossivulla Käyttäjän lisätiedot- painiketta.



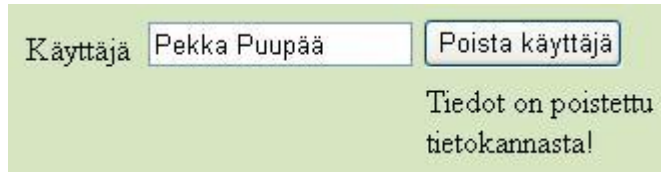
Oikealle avautuvalla lisätietosivulla painetaan käyttäjänimen oikealla puolella olevaa Poista käyttäjä- painiketta



jonka jälkeen ilmestyy varmistusikkuna, jossa kysytään hyväksyntää tietojen poistamiselle.



Mikäli painetaan Peruuta-painiketta, tietoja ei tällöin poisteta järjestelmästä. Jos painetaan OK-painiketta, tiedot poistetaan järjestelmästä ja Poista käyttäjä-painikkeen alla näytetään tästä varmistusviesti.

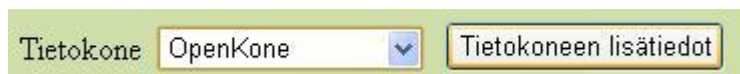


Käyttäjä Pekka Puupää Poista käyttäjä

Tiedot on poistettu tietokannasta!

This is a screenshot of a web application interface. It features a light green background. At the top, there is a label 'Käyttäjä' followed by a text input field containing 'Pekka Puupää'. To the right of the input field is a button labeled 'Poista käyttäjä'. Below these elements, centered, is a confirmation message: 'Tiedot on poistettu tietokannasta!'.

Laitteita poistetaan suorittamalla ensin tilanimen perusteella tehty haku. Mikäli poistettava laite on jokin muu kuin videotykki tai dvd/vhs-laite, valitaan poistettava laite alasvetovalikosta ja siirrytään lisätietosivulle lisätietopainiketta painamalla.



Tietokone OpenKone ▼ Tietokoneen lisätiedot

This is a screenshot of a web application interface. It shows a label 'Tietokone' followed by a dropdown menu with 'OpenKone' selected and a downward arrow. To the right of the dropdown is a button labeled 'Tietokoneen lisätiedot'.

Mikäli poistettava laite on tietokone, Poista tietokone- painike sijaitsee lisätietosivulla tietokoneen nimen vieressä.

Tietokone	<input type="text" value="OpenKone"/>	<input type="button" value="Poista tietokone"/>
Ip	<input type="text" value="192.168.1.20"/>	
Malli	<input type="text" value="Osborne"/>	
Käyttöjärjestelmä	<input type="text" value="Win 7"/>	
Serial	<input type="text"/>	
Leasing	<input type="text"/>	
Cpu	<input type="text"/>	
Ram	<input type="text"/>	
Kovalevy	<input type="text" value="Maxtor 400 Gb"/>	
Tila	<input type="text" value="Luokka1"/>	
<input type="button" value="Tallenna tietokoneen muutokset"/>		

Jos kyseessä on jokin muu laite kuin tietokone, kulloisenkin laitteen mallin vieressä on poistopainike.

Ip	<input type="text" value="4.5.6.7"/>	
Malli	<input type="text" value="Brother"/>	<input type="button" value="Poista tulostin"/>
Serial	<input type="text" value="0"/>	
Kasetti	<input type="text" value="0"/>	
Tila	<input type="text" value="Luokka1"/>	
<input type="button" value="Tallenna tulostimen muutokset"/>		

Laitteen poistamisen yhteydessä kysytään varmistusta, kuten käyttäjänkin poiston yhteydessä.

## 8. TIETOJEN MUOKKAAMINEN

Tietoja voi muokata suorittamalla ensin käyttäjä- tai tilanimen perustuvan haun. Tietoja pääsee muokkaamaan tarkastelemalla laitteen/käyttäjän lisätietoja ja syöttämällä päivitettävät tiedot suoraan vanhan tiedon päälle lisätietosivulla. Esimerkkinä on tulostimen tietojen muokkaaminen.

Tila	<input type="text" value="Luokka1"/>	
Käyttäjä	<input type="text" value=""/>	
Tietokone	<input type="text" value="OpenKone"/>	<input type="button" value="Tietokoneen lisätiedot"/>
Tulostin	<input type="text" value="Brother"/>	<input type="button" value="Tulostimen lisätiedot"/>
Näyttö	<input type="text" value="Samsung"/>	<input type="button" value="Näytön lisätiedot"/>
Videotykki	<input type="text" value="3M"/>	<input type="button" value="Videotykin lisätiedot"/>
Dvd/Vhs	<input type="text" value="LG"/>	<input type="button" value="Dvd/Vhs:n lisätiedot"/>
Äänentoisto	<input type="text" value="Kuulokkeet"/>	

Ip	<input type="text" value="4.5.6.7"/>	
Malli	<input type="text" value="Brother"/>	<input type="button" value="Poista tulostin"/>
Serial	<input type="text" value="0"/>	
Kasetti	<input type="text" value="0"/>	
Tila	<input type="text" value="Luokka1"/>	
<input type="button" value="Tallenna tulostimen muutokset"/>		

Ip 4.5.6.7

Malli Brother Poista tulostin

Serial 30fk7hd

Kasetti 0

Tila Luokka1

Tallenna tulostimen muutokset

Muokattujen tietojen tallennus tapahtuu painamalla Tallenna tulostimen muutokset- painiketta.

Onnistuneen tallennuksen jälkeen tallennuspainikkeen alapuolella tulostuu teksti ”Muutokset on tallennettu tietokantaan!”.

Tallenna tulostimen muutokset

Muutokset on tallennettu  
tietokantaan!